c't
magazin für computer technik

5/84

DM 6,—
öS 52,—
sfr 6,—
hfl 6,80

c't-Serie
BASIC intern
★ Was nicht im Handbuch steht ★

SuperTape für C 64 und VC-20

Den MACRO-80 meistern

c't-Floppy-Karte in ECB-Systemen

SBASIC-Preprozessor

Polygon-Berechnung

Prüfstand:

MTX 512

ORIC Atmos

Tandy Colour 2

Fehlersuche leicht gemacht

c't — ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 2746, 3000 Hannover 1

5 April/Mai 1984

Heise Software-Service

Ein Extra-Service des Heise-Software-Service ◄◄◄◄◄◄◄◄◄◄◄◄

VICTAPE COMPUTING für VC-20

SPECTRUM COMPUTING für ZX-Spectrum

MODEL B COMPUTING für BBC-Acorn, Modell B

Spezial-Magazine auf Compact-Kassetten

COMPUTING demonstriert, was Ihr Homecomputer kann:

★ Systemprogramme ★ Spiele
★ Bewegliche Grafik ★ Internationale News

auf dem Farbbildschirm

COMPUTING-Magazine — exklusiv für c't-Leser
in der internationalen Originalausgabe (in englischer Sprache)

Preis: 19,80 DM (zuzüglich 3 DM Versandkosten)

Für Sinclair-Fans:

ZX-COMPUTING

das große Spezialmagazin für ZX81 und Spectrum
(Originalausgabe in englischer Sprache)

Programme und Informationen
über Ihren Computer

Preis: 9,80 DM (zuzüglich 1,70 DM Versandkosten)

Nummer 4

Nummer 5

Bestellen beim

Heise-Software-Service
Postfach 27 46 · 3000 Hannover 1

Lieferung nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck oder den quittierten Einzahlungsbeleg Ihrer Bank bei. Überweisungen bitte auf das Konto-Nr. 93 05-308 Postscheckamt Hannover

c't-Abonnement

Abrufkarte

GARANTIE

Wir garantieren jedem Abonnenten das Recht, seine Bestellung innerhalb einer Woche nach Abschluß schriftlich zu widerrufen.

c't-Abonnement Abrufkarte

Ja, übersenden Sie mir bis auf Widerruf alle künftigen c't-Ausgaben ab Monat

_____

(Kündigung 8 Wochen zum Jahresende möglich.)

Das Jahresabonnement kostet DM 58,— inkl. Versandkosten und MwSt.

Absender und Lieferanschrift
Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Straße/Nr.

PLZ Wohnort

_____

Datum/Unterschrift

Ich bestätige ausdrücklich, vom Recht des schriftlichen Widerrufs innerhalb einer Woche nach Abschluß beim Verlag Heinz Heise GmbH, Postfach 27 46, 3000 Hannover 1, Kenntnis genommen zu haben.

Unterschrift _____
Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

c't-Kontaktkarte

Mit dieser Service-Karte können Sie

- **Informationen** zu in **c't** besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierenden oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Folien, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, **c't**-Versand, Postfach 27 46, 3000 Hannover 1, **ordern**.

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in **c't** _____/8__, Seite _____ erschienene
☐ Anzeige ☐ redaktionelle Besprechung
☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____
☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't-Kontaktkarte

Mit dieser Service-Karte können Sie

- **Informationen** zu in **c't** besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierenden oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Folien, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, **c't**-Versand, Postfach 27 46, 3000 Hannover 1, **ordern**.

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in **c't** _____/8__, Seite _____ erschienene
☐ Anzeige ☐ redaktionelle Besprechung
☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____
☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't-Abonnement **Abrufkarte**

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleitzahl Konto-Nr.

Geldinstitut Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Antwortkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

magazin für computer technik

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

c't-Abonnement
Abrufkarte

Abgesandt am

_____ 198__

zur Lieferung ab

Heft _____ 198__

Jahresbezug DM 58,—
inkl. Versandkosten und MwSt.

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen. ▶

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

_____ 198__

an Firma _____

Bestellt/angefordert

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen. ▶

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

_____ 198__

an Firma _____

Bestellt/angefordert

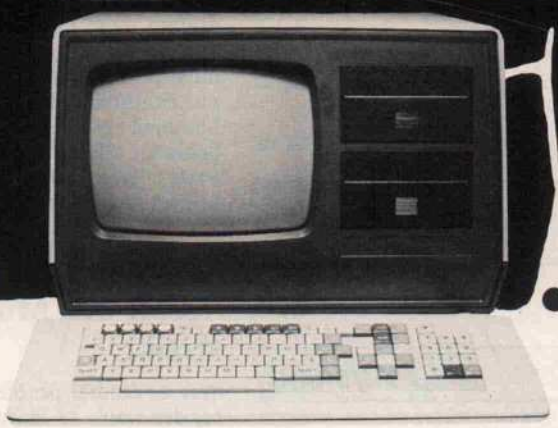
nascom-C

CP/M ist ein eingetragenes Warenzeichen der Firma Digital Research

Jumbo-Komfort für Ihren Tischrechner

bietet Ihnen das neue CP/M 3.0 Betriebssystem!

NASCOM-C ist maßgeschneidert für CP/M3



Kompatibilität und Systemkomfort.
NASCOM-C ist ein neues deutsches System zur bestmöglichen und komfortabelsten Ausnutzung der modernsten CP/M-Software-Produkte.

Ergonomie für jedermann erschwinglich.
Ein System, das einiges auf dem Kasten hat und trotzdem viel preiswerter ist als so mancher Homecomputer, vor allem voll ausgebaut. Es entspricht den deutschen Vorstellungen von Benutzerfreundlichkeit, Service und Ergonomie. Ob Sekretärin, Handwerker, Student, Tüftler, Techniker oder Manager: Jeder findet in NASCOM-C einen zuverlässigen Freund, der allzeit bereit nervtötende Routinearbeit übernimmt und seinen menschlichen Partner versteht anstatt ihn zu veräppeln.

Ein System, das mitdenkt.
NASCOM-C versteht Sie, auch wenn Sie bisher nichts über Computer wissen, hilft Ihnen aber auch, Computer zu verstehen, hat keine Geheimnisse. Das geht so weit, daß Sie ihn auch selbst zusammenbauen und so von Grund auf kennenlernen können. Wer seinem NASCOM-C ein bißchen Zeit widmet, kann so viel Geld sparen und noch mehr lernen.

Gute Software braucht bessere Hardware!
NASCOM-C bedient sich des neuen CP/M-Plus-Betriebssystems, das bisher auf Mikros nicht bekannten Komfort mit der Kompatibilität zu seinen Vorläufern verbindet, für welche das größte Angebot an hochwertiger Software existiert. Aber was nützt die schönste Software, wenn die vorhandene Hardware deren Möglichkeiten nicht nutzen kann? Faule Kompromisse auf der Hardwareseite sind bei CP/M+ nämlich schwerlich möglich. Bessere Betriebssysteme brauchen nun mal modernere Computer. NASCOM-C ist ein für CP/M+ maßgeschneiderter Rechner, der allen alle Möglichkeiten von CP/M+ bietet, ohne Nerven, Geldbeutel oder Spezialkenntnisse zu überfordern.

Wer bietet mehr auf einer Karte?
- Z80 A/B Zentraleinheit mit 4/6 MHz Takt
- Speicherverwaltungseinheit (MMU) und DMA
- 128 Kilobyte Arbeitsspeicher mit Paritätsprüfung, erweiterbar auf 1 Megabyte
- Floppy-Disk Controller für alle 5¼- und 8-Zoll Laufwerke

- Festplattenschnittstelle vorhanden
- Ein DEC VT52, -HEATH H19 -und ANSII-Aufwärtskompatibles Terminal mit Grafik in 8 Farben und ladbarem Zeichengenerator
- Zwei V.24 und eine Centronics-Schnittstelle zum Anschluß von Druckern, Plottern, Modems und Hostrechnern
- Über RS-422 Schnittstellen zum Netzwerk erweiterbar
- Der 77-polige NASBUS macht NASCOM-C kompatibel zu vielen Erweiterungskarten (wie Farbgraphik mit 792*256 Punkten und 256 KB Speichererweiterungen
- Farbgraphik kompatibel zu den Normen Tektronix 40XX, Plot-10 und GKS, umfangreicht Objektcode-Bibliotheken
- Jede wichtige Programmiersprache verfügbar
- Branchenlösungen, Text- und Datenbanksysteme in kaum überschaubarer Vielfalt
- Von der Leerplatine mit Dokumentation und Firmware bis zum Fertiggerät lieferbar
- Der Grundbausatz einschließlich Dokumentation und Firmware kostet unter DM 1.000,–, als Leerplatine sogar nur DM 298,–!!

Wer lieber gleich den richtigen Computer kaufen will oder den Frust mit seinem jetzigen satt hat, bekommt für 2 DM sein NASCOM-C INFO-Paket direkt von

LAMPSON & ZERBE GmbH
Mikrocomputervertrieb
Odenwaldstraße 21–23
6087 Büttelborn · Tel. 0 61 52 / 5 67 30

Inhaltsve

c't 5/84

8 **Leserbriefe**

14 **c't aktuell**

50 **c't-Club**

102 **c't-Software-Review**

104 **c't-Buchkritik**

105 **Platinen-Service**

111 **Software-Service**

107 **Inserenten-Verzeichnis**

108 **Vorschau auf Heft 6/84**

108 **Impressum**

c't-Prüfstand

26 **MTX 512: Ein Musterknabe**
Der lang Erwartete von Memotech

48 **Kein großer Wurf**
Tandy TRS 80 Colour Computer 2

64 **Der neue ORIC**
ATMOS auf dem Prüfstand

c't-Praxistips

31 **Wenn der Mikro streikt**
Triggerprogramme für die Fehlersuche mit dem Oszilloskop

61 **ORIC — ROM geknackt**
Teil 1: Speicheraufbau und Systemadressen

63 **Laden ohne Starten**
Ein CP/M-Tip

77 **c't-Floppy-Interface**
in 'gewöhnlichen' ECB-Systemen

98 **Der verflixte INPUT**
beim CBM 8032

c't-Titel

Scope-Extender

Jeder, der sich mit Computer-Hardware ernsthaft befaßt, wird sehr bald mit dem Problem des 'timings' logischer Abläufe konfrontiert. Ein Oszilloskop wird rasch zum unentbehrlichen Meßinstrument. Allerdings ist es sehr mühsam, mit Hilfe eines Zwei- oder gar Einkanalers, ein Timing-Diagramm zu erstellen. Unser Scope-Extender kommt Ihnen da sehr entgegen: Sie brauchen nur ein Einkanal-Scope und bekommen 16 Signale auf den Schirm. Gleichzeitig! — Der Preis? Weit unter 20 000 D-Mark! Etwa so bei 60.

Seite 36

Wenn der Mikro streikt

Wer es einmal probieren mußte, der weiß: so einfach ist die Fehlersuche an Mikroprozessorsystemen mit dem Oszilloskop gar nicht. Ein undurchschaubares Wirrwarr auf dem Bildschirm dürfte der Regelfall sein. Wir zeigen Ihnen, wie man sich mit Hilfe von Triggerprogrammen systematisch durch ein funktionsunwilliges Prozessorsystem zum Fehler hin 'durchmißt'. Dabei ist es gleichgültig, ob es sich nur um einen Rechnerausfall handelt, oder ob eine Neuentwicklung zum Laufen gebracht werden soll.

Seite 31

Centronics-Interface für den Spectrum 16/48K

Langsam wird unsere Reihe der Centronics-Interfaces komplett und damit der Weg frei zur Textverarbeitung ohne 'Silberstreif'. Außerdem ist nun auch der Spectrum in die erlesene Schar der Computer aufgenommen, mit denen man den c't-Sprachsynthesizer betreiben kann.

Seite 29

SuperTape für 6502-Computer

Mit der Kompatibilität zwischen VC-20 und C64 hapert's ein wenig, wie jeder weiß, der es einmal mit dem Software-Tausch per Kassette versucht hat. SuperTape löst das Problem. 'Nebenbei' wird auch die Chance zur Kommunikation mit anderen Rechnern geschaffen, die SuperTape kennen. Und das alles bis zu sechsmal schneller als bisher. Na bitte!

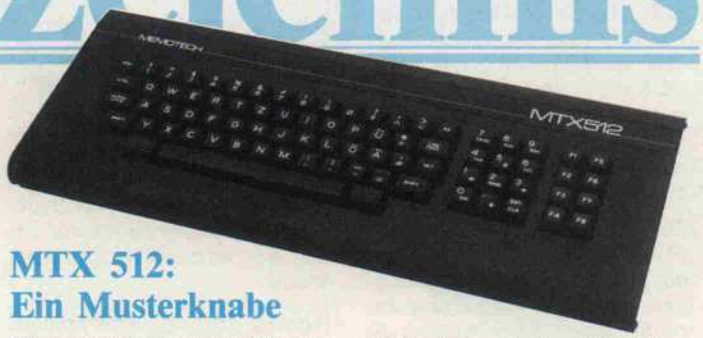
Seite 87

Der neue ORIC ATMOS

Bei näherem Hinsehen kam er uns bekannt vor: Der neue ORIC ATMOS ist ein ORIC 1 in neuem Gewand. Den Schritt zu einer 'richtigen' ASCII-Tastatur hat der Hersteller nun doch gewagt. Auch das Betriebssystem wurde verbessert. Einerseits wurden einige Fehler daraus entfernt, zum andern sind ein paar hübsche Eigenschaften hinzugekommen.

Seite 64

rzeichnis



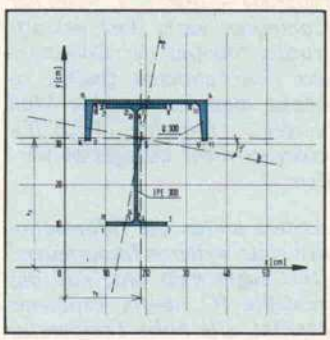
MTX 512: Ein Musterknabe

Ein neuer Computer im 'Tastatur-Look'. Mit vielen Features, die Commodore in dieser Gattung als Maßstab gesetzt hat. Aber Gott sei Dank nicht so antikompatibel zu 'nicht typengebundenen' Peripherie-Geräten. Und er ist nach oben hin offen: der MTX kann zu einem CP/M-fähigen Rechner ausgebaut werden. Das hat allerdings seinen Preis.

Seite 26

Polygone

... in Deutschland oft auch entmystifiziert als 'Vielecke' bezeichnet, bergen, wenn sie nur drei bis vier Ecken haben, keine großen Berechnungsprobleme. Auch der Flächeninhalt von Elfecken läßt sich mit Bleistift und Radiergummi noch rausbekommen. Wenn dann aber noch Fehlflächen ins Spiel



kommen und jemand von Ihnen die Ermittlung der Trägheitsmomente, bezogen auf die Schwerachsen eines Profils, das nirgends tabelliert ist, verlangt, dann werden Sie vermutlich von unserem Programm hellauf begeistert sein. Es kann sogar noch mehr!

Seite 80

Wenn 64K nicht reichen

Bedeuteten dem Computerbesitzer noch vor gar nicht langer Zeit Speichererweiterungen um 1 KByte schon einen gewaltigen Fortschritt, so gibt man sich heutzutage vielfach mit 64 K nicht mehr zufrieden. Wenn alle ROMs abgeklemmt und durch RAMs ersetzt sind, dann geht das Tüfteln los. Wir beleuchten verschiedene Verfahren des Speicherausbaues, vom virtuellen Speicher bis zur RAM-Floppy.

Seite 52

MACRO-80 meistern

Der 'M80' ist weit verbreitet und doch meistens sehr schlecht dokumentiert, da er vielfach nur Compilern der Firma Microsoft 'beigefügt' wird. Wer sein eigenes und das Können dieses Assemblers der Spitzenklasse ausschöpfen möchte, der wird in diesem Beitrag auf den Geschmack gebracht. An kleinen Beispielprogrammen, die der Leser ausprobieren kann, werden die entscheidenden Fähigkeiten des M80 demonstriert.

Seite 68

c't erscheint 12mal im Jahr zur Monatsmitte.
c't 6/84 erhalten Sie ab **17. 5. 84.**

Software-Know-how

Arithmetik-Unterricht für 6502 und Z80
Teil 3: BCD-Arithmetik — **40**

BASIC intern
Was nicht im Handbuch steht, Teil 2 — **45**

FORTH anpassen
Teil 3: Der Massenspeicher — **66**

MACRO-80 meistern
Ein Software-Werkzeug mit allen Schikanen, Teil 1 — **68**

c't-Programme

SBASIC-Preprozessor
Strukturiert programmieren in Commodore-BASIC — **56**

Polygone
Vom Umfang bis zu den Flächenmomenten — **80**

c't-Projekte

Centronics-Interface
für ZX-Spectrum mit 16 und 48 K — **29**

Scope-Extender
Digitales Vorschaltgerät bringt 16 Kanäle auf den Bildschirm — **36**

c't-Terminal 'nach Maß'
Teil 3: Korrekturen und Hinweise für die Programmanpassung — **85**

SuperTape für 6502-Computer
Beispielhaft vorgeführt am VC-20 und C64 — **87**

SuperTape-Komfort
für den ZX 81 — **92**

c't 86 — Zwischenbilanz
Anmerkungen und Berichtigungen — **96**

c't-Applikation

VIA 6522
Einsatz und Programmierung — **75**

Grundlagen

Wenn 64 K nicht reichen
Banking, Paging, virtueller Speicher — **52**

Schaltbilder zu 'mager'

Vor einigen Tagen fielen mir zufällig die ersten drei Hefte Ihrer Zeitschrift c't in die Hände. Zum Gesamtkonzept kann ich Ihnen nur gratulieren. Machen Sie bitte so weiter.

Bei allem Lob trotzdem noch ein Verbesserungsvorschlag:

Die Schaltbilder sollten meiner Ansicht nach vollständiger beschriftet werden. Beim c't 86 fiel mir das Verständnis der Schaltung wegen fehlender IC-Pin-Funktionsbezeichnungen recht schwer. Ich hätte zum Beispiel beim Floppy-Controller an den IC-Anschlüssen gerne Bezeichnungen, wie sie im Original WD-Datenblatt verwendet werden, vorgefunden. Dadurch wäre denjenigen Lesern, die Ihre Schaltungen verstehen möchten, einiges an mühseligem Datenblatt-Studium erspart geblieben.

Dipl.-Ing. R. Willer, Wedel

Wir wollen uns bessern. (Red.)

8-Bit nicht vergessen

In Ihrer Zeitschrift c't Heft 3, 1984, berichten Sie über die Möglichkeit, beim VC-20, ZX81 und ZX Spectrum mehr Speicherplatz zu erhalten. Ihr Bemühen darum ist gleichwohl lobenswert wie wenig außergewöhnlich, da sich viele, wenn nicht alle Computerzeitschriften mit Computern dieser Preisklasse beschäftigen. Sehr selten dagegen erhält man derartige Informationen für Rechner der mittleren und höheren Preisklasse. Ich erwarte nun nicht, daß sich jede Zeitschrift ausschließlich mit letzteren Maschinen beschäftigt, doch sollten, vielleicht entsprechend dem Marktanteil, gelegentlich auch für diese Computer einige Druckseiten zur Verfügung stehen.

Es ist gut, daß Sie sich für die Entwicklung eines 16-Bit-Rechners einsetzen, aber vergessen Sie nicht die vielen Anwender, die Tausende DM in ihr 8-Bit-System gesteckt haben und nun nach und nach dieses 8-Bit-System vervollständigen wollen. Vernachlässigen Sie daher nicht eigene Entwicklungen im Bereich der 8-Bit-Systeme.

Romanus Wille, Bochum

(c't 3/84, Brief von H.-G. Otto)

Noch mehr Speicherraum

Wem der 1 MByte Adreßraum des 8086 noch zu klein sein sollte, nehme doch einfach eine (oder mehrere) Portleitungen zu Hilfe und verwende diese als zusätzliche Adreßleitungen. So können Sie dann zwischen nahezu unbegrenzt vielen 1 MByte Speicherbänken umschalten. Eine kleine Zusatzschaltung muß jedoch gewährleisten, daß Sie beim Umschalten keine Systemadressen, den Stack usw., ausblenden.

Ferner möchte ich darauf hinweisen, daß ein 8086-System relativ leicht zu einem Multiprozessor-System erweitert werden kann. Dies erhöht die Rechenleistung erheblich, so daß sich der 8086 gegenüber vielen Konkurrenten abhebt.

Klaus Scheller, Neubiberg

FORTH-Fragen

Die FORTH und GraFORTH Artikel von Peter Glasmacher haben mich angeregt, selbst FORTH zu lernen und einzusetzen. Dazu einige Fragen:

Der FORTH von c't Programmbibliothek Nr. 8 enthält u.a. Zeilen-Editor, Decompiler und Textformattierer. Inwiefern sind diese beim normalen FORTH Listing enthalten?

Ich habe gelesen, ('Starting FORTH', Leo Brodie), daß FORTH zugleich Compiler, High-Level Sprache und Betriebssystem ist. Ist das immer der Fall, oder läuft z. B. das c't FORTH unter Apple-DOS (ich denke an den Bericht über Jupiter Ace)?

Gibt es vielleicht im Raum Bremen noch jemanden, der sich für FORTH interessiert? Ich wäre dankbar für die Anschrift(en) bzw. Weitergabe meiner Adresse mit der Absicht zur gegenseitigen Unterstützung.

Gibt es eine FORTH Interessengemeinschaft (FIG?!) in der BRD?

Ansonsten danke ich für eine interessante neue Zeitschrift und hätte gerne noch mehr FORTH und IBM-PC Informationen in c't!

Eugen Klaußner, Lilienthal

Die zusätzlichen Programme, die den diversen FORTH-Systemen beigefügt sind, richten sich meist nach dem 'Gefühl' des Entwicklers und dem Preis. Standard ist lediglich der Zeileneditor (siehe 'STARTING FORTH') und bei 65xx-Zielprozessoren ein Assembler (soweit bekannt, gibt es nur bei AFORTH einen 65C02 Assembler.)

Die Aussage in Starting FORTH ist insoweit korrekt, als FORTH auf einem beliebigen Prozessor seine eigene hypothetische CPU simuliert. Es werden, grob gesagt, FORTH-spezifische Instruktionen kompiliert (COMPILE), welche von einem in der jeweiligen Maschinensprache geschriebenen winzigen 'Interpreter' analysiert und ausgeführt werden (NEXT). Dieser Interpreter umfaßt je nach Prozessor 9 bis 25 Bytes.

Ob im Raum Bremen noch weitere FORTH-Fans existieren, ist uns momentan nicht bekannt, vielleicht melden sich einige auf diese Anfrage.

Eine Sektion der FORTH-Interest-Group existiert auch in der Bundesrepublik (siehe c't-Club in dieser Ausgabe).

(Red.)

HX 20 'knacken'?

Gestatten Sie mir als eifrigem Leser und Abonnent Ihres sehr interessanten c't-Magazins folgende Anfragen in bezug auf den HX 20-Computer:

Ließe sich das in Ihrem Heft 4/1984 vorgestellte SuperTape-Verfahren auch beim HX 20 anwenden? Und ist beabsichtigt, demnächst auch eine diesbezügliche Version vorzustellen?

Als Handicap für Maschinensprache-Programme erscheint mir das Geheimnis um die interne Software. Für das kommende Heft kündigen Sie einen Beitrag über das geknackte ORIC-ROM an. Darf man anregen, daß Sie sich einmal diesbezüglich des HX 20 annehmen?

Prof. Dr. K. H. Kreeb, Worpswede

Eine Ankündigung im Hinblick auf die gewünschten Themen können wir leider nicht machen, denn auch bei uns fehlt es derzeit noch an den nötigen Detailkenntnissen über den HX 20. Wir hoffen aber auf Beiträge aus dem Leserkreis.

(Red.)

ZX 81 im 'Dauerstand'

Ich besitze seit einem Jahr einen ZX 81 mit 16K, der bis vor kurzem einwandfrei funktionierte, dann aber für immer in den 'Dauerstand' überging, auch nach dem Einschalten. Bei geöffnetem Gehäuse merkt man, daß die ULA (das 'Kundenchip') heißer als normal ist, da sie vermutlich überlastet wurde.

Ähnlich geht es einem Freund, aber ich denke, wir sind nicht die einzigen mit diesem Problem. Daher meine Fragen: Woher kann ich eine ULA bekommen, und wie könnte ich verhindern, daß dieser Baustein wiederholt zerstört wird?

Wolfram Nücker, Bruchsal

Wie beim Sinclair-Generalimporteur zu erfahren war, sind dort keine Probleme mit der ULA in größerem Umfang bekannt geworden. Zwar soll es 'normal' sein, daß dieses IC bis zu 180° C! heiß wird, die Ausfallquote liege aber im Bereich des normalen Verschleißes. Wenn trotz allem ein ZX81 ausfällt, wird er bei der Firma repariert — aber nur, wenn der Computer auch dort gekauft wurde. Stammt der Computer aus einer anderen Quelle, so könne man ihn 'in den Müll werfen'. Einzelne ULA-ICs könnten nicht abgegeben werden.

Anders lauten die Informationen eines weiteren Importeuers. Dort sagte man uns, daß das bewußte IC häufig Probleme bereitet. Die hohe Temperatur des Bausteins sei konzeptbedingt und die Ursache für Ausfälle. Nach eigenen Messungen der Firma soll der 'Tod' bei circa 72 °C im Rechnergehäuse eintreten. Da von der erstgenannten Firma keine Ersatzteile zu erhalten seien, wird bei einem zur Reparatur eingesandten Gerät im Falle eines ULA-Defekts die gesamte Platine ausgetauscht. Das geschieht zu einem Festpreis — auch bei Geräten, die nicht 'im Hause' gekauft wurden.

Wir erlauben uns den Hinweis, daß die üblicherweise zulässige Sperrschichttemperatur bei Siliziumchips bei maximal 150 °C liegt. Um die Temperatur im Gehäuse des ZX zu verringern, sollten Sie das Kühlblech des Stabi-ICs vergrößern. (Red.)

Was ist Ihnen lieber? Ein Drucker mit diversen Aufpreis-Extras oder ein MC 2100?

Nur DM 1820,– incl. MwSt. kostet Sie (bei Einzelstück-Abnahme) der neue Schönschrift-Matrixdrucker MC 2100. In diesem Preis ist all das enthalten, das andere nur gegen Aufpreis bieten können.
Mehr sollten Sie nicht bezahlen – weniger Leistung dürfen Sie von Ihrem künftigen Drucker nicht verlangen.

Wir suchen Vertriebspartner, die auch im Druckergeschäft noch was verdienen wollen.

Dyneer

Technitron GmbH
eine Firma der Dyneer Gruppe

Charles-de-Gaulle-Straße 4
8000 München 83
Tel. (089) 637 30 90, Tlx. 05 22 585

Büro Hamburg: Tonndorfer Hauptstraße 126, 2000 Hamburg 70, Tel. (040) 66 91 81, Telex 2174314
Büro Düsseldorf: Kaarster Straße 18, 4005 Meerbusch 2, Tel. (0 21 59) 40 41/42

c't 86

Als allererstes möchte ich Ihre Idee, einen 16-Bit-Mikrocomputer für den Selbstbau vorzustellen, loben. Nachdem sich unzählige Firmen auf dem 8-Bit-Selbst- beziehungsweise Teilselbstbau-Markt versucht haben, war es an der Zeit, dem engagierten Hobbyisten die Möglichkeit zu geben, sich ein echtes 16-Bit-System zu erstellen.

An der vorgestellten Hardware sind mir aber drei Punkte negativ aufgefallen:

Der Prozessor 8086 adressiert meines Wissens 1 MByte RAM, warum sehen Sie nur drei Speicherkarten mit insgesamt 768 KByte vor?

Auf dem Systembus des 8086 stehen alle notwendigen Steuersignale für einen 'hidden refresh' des RAM zur Verfügung. Warum verwenden Sie die Uralt-Lösung mittels einer 'refresh'-Routine, die zusätzliche Rechnerzeit kostet und damit das System unnötig verlangsamt?

Wenn Sie sich schon soviel Mühe mit der Einprogrammierung eines walzenförmigen Bildwiederholungsspeichers gegeben haben, frage ich mich, warum Sie eine so geringe Seitenzahl, nämlich zwei, vorgesehen haben. Die Preise für RAM sind heutzutage soweit gefallen, daß ich eine höhere Seitenzahl, zum Beispiel acht Seiten, für durchaus angebracht halte, zumal durch diese Entwicklung die Programme durchschnittlich länger geworden sind und so eine größere Übersichtlichkeit, z. B. des Listings, wünschenswert wäre. (Diese Aussage bezieht sich offenbar auf das c't-Terminal. Red.)

Ist eine Controller-Karte für die neuen 3,5-Zoll-Disketten vorgesehen? Wie ist es um eine Grafik-Karte bestellt? Welche Auflösung hat sie? Wird sie von der Betriebssoftware unterstützt? Läuft der Computer auch mit einem anderen, als mit dem von Ihnen vorgestellten Selbstbau-Terminal?

Wie ist es um die Flexibilität der Betriebssoftware bestellt? Ist es zum Beispiel möglich, ohne große Mühe, eigene Befehle zu definieren und sie in den 'Grundwortschatz' des Computers aufzunehmen? Ist das System MP/M-fähig? Wie sieht es mit der Software-Kompatibilität zu anderen Computern aus? Ist es möglich, *jedes* unter CP/M erstellte Programm ordnungsgemäß auf dem Computer ablaufen zu lassen, gesetzt den Fall, daß die nötige Hardware zur Verfügung steht?

Wolfram Herkendell, Bonn

Beim c't 86 sind 'nur' 768 KByte RAM vorgesehen, da das EPROM nicht ausgeblendet wird und ein Speicherbereich für Video-Grafik vorgesehen ist.

Die verwendete Refreshroutine benötigt maximal 2% der Rechnerzeit. Außerdem sind nicht alle Speicher-ICs für einen hidden refresh geeignet. Die CPU kann mit einer maximalen Taktfrequenz von 8 MHz laufen (theoretisch). Die hidden-refresh-Technik funktioniert nur bis zu einer maximalen Taktfrequenz von 5 MHz.

Vermutlich beziehen Sie sich mit Ihrer Frage nach der Größe des Bildwiederholspeichers auf den c't-Terminal-Computer. Die meisten Video-Interfaces weisen nur eine Bildseite auf, das c't-Terminal bietet immerhin das Doppelte. Wir meinten, damit einen akzeptablen Kompromiß gefunden zu haben.

Eine Controller-Karte für 3½"-Laufwerke ist nicht vorgesehen (es gibt aber Laufwerke, die 5¼"-kompatibel sind), eine Grafik-Karte ist in Vorbereitung.

Der c't 86 kann mit jedem Terminal arbeiten, das über eine serielle Schnittstelle verfügt.

Die Definition eigener Befehle ist nur dann möglich, wenn Sie einen Sprachcompiler (z. B. FORTH) benutzen, der die Definition erlaubt. Eigene Maschinenbefehle können Sie nicht definieren.

Der c't 86 ist MP/M-fähig und bedingt softwarekompatibel zum IBM-PC. Wenn der c't 86 mit der nötigen Hardware ausgerüstet ist, kann jedes unter CP/M erstellte Programm auf ihm laufen. *(Red.)*

Bessere Label-Lösung

Ich möchte einige Anmerkungen zu Heinz-Peter Heidingers Artikel '... aber bitte mit Labels!' in Heft 3/84 Ihrer Zeitschrift machen. Die Idee, RESTORE n nicht mit Zeilennummern, sondern mit Labels zu realisieren, finde ich sehr gut. Leider ist die vorgestellte Lösung nicht ohne 'Macken'.

Label, die Basic-Worte enthalten, werden nicht gefunden. Der Grund dafür ist, daß in der Referenz-Zeile eventuelle Basic-Worte nach der Eingabe der Zeile in Token umgewandelt werden. In der Label-Zeile geschieht dies wegen des Remark-Befehls (') nicht.

Kleine Buchstaben können in Labels nicht verwendet werden. In der Referenz-Zeile werden sie in Großbuchstaben umgewandelt, in der Label-Zeile nicht.

Unter bestimmten Bedingungen werden falsche Label-Zeilen gefunden. Ist in der Referenz-Zeile beispielsweise 'RSET TO DAT11' angegeben und gibt es im Programm eine Label-Zeile "DAT1' vor der Label-Zeile "DAT11', so wird der DATA-Zeiger hinter "DAT1' gesetzt. Der Grund dafür ist, daß nur in der Label-Zeile das Ende des Labels geprüft wird, nicht aber in der Referenz-Zeile.

Diese Probleme treten nicht mehr auf, wenn die Syntax für die Referenz-Zeile geändert wird zu 'RSET TO "Label"'. Zum einen wird durch die Anführungszeichen das Label nach der Eingabe der Zeile als String betrachtet. Eventuell darin enthaltende Basic-Worte werden dann nicht in Token umgewandelt. Ebenso findet keine Umwandlung von Kleinbuchstaben in Großbuchstaben statt. Zum anderen dient das hintere Anführungszeichen zur Erkennung des Label-Ende. Für die neue Syntax der Referenz-Zeile sind folgende Änderungen im Assembler-Programm notwendig:

Zeilen 18—92 bleiben unverändert.

Zwischen Zeile 92 und 93 einfügen:

```
CP '"'; Anführungszeichen
        vor Label?
JP NZ, 1997H; nein: SN-
        Error melden.
INC HL; ja: Zeiger auf 1.
        Label-Zeichen.
```

Zeilen 93—168 bleiben unverändert.

Zeile 169 ändern:

```
169 JR Z, REFEND; ja: prüfen,
        ob Label in Referenz-
        Zeile zu Ende.
```

Zeilen 170—174 bleiben unverändert.

Zwischen Zeile 174 und 175 einfügen:

```
REFEND LD A, (HL); Ist
        nächstes Zeichen in Refe-
        renz-Zeile ein Anführungs-
        zeichen?
CP '"'
RET; ja: Z-Cond. — Label
        gefunden;
nein: NZ-Cond. — weiter-
        suchen.
```

Gerd Kluge, Witten

COMPACT-Interface

Mit Freude stellte ich fest, daß Sie als erste Computer-Zeitschrift endlich ein Interface für meine Schreibmaschine veröffentlicht haben.

Allerdings stellt sich bei dem sonst hervorragenden Konzept ein kleines Problem: Ich habe nicht die Möglichkeit, den in diesem Interface eingesetzten Prozessor 8748 zu programmieren. Nun möchte ich anfragen, zu welchem Preis Sie mir einen bereits fertig programmierten Prozessor liefern könnten.

Reiner Buch, Berlin

Wo kann ich weitere Informationen über den 8748 beziehen (Datenblatt etc.)?

Jürgen Blum, Bad Soden

Der programmierte Prozessor ist beim Autor des Artikels erhältlich. Hier die Anschrift: Herbert Nabereit, Köterei 18 B, 3300 Braunschweig. Leider ist der IC zur Zeit knapp und teuer. Aus diesem Grund haben wir vorläufig davon abgesehen, es in den Software-Service aufzunehmen.

Informationen und Datenblätter sind bei der Firma Intel erhältlich. Die Anschrift ist: Intel Semiconductor GmbH, Seidlstraße 27, 8000 München 2. *(Red.)*

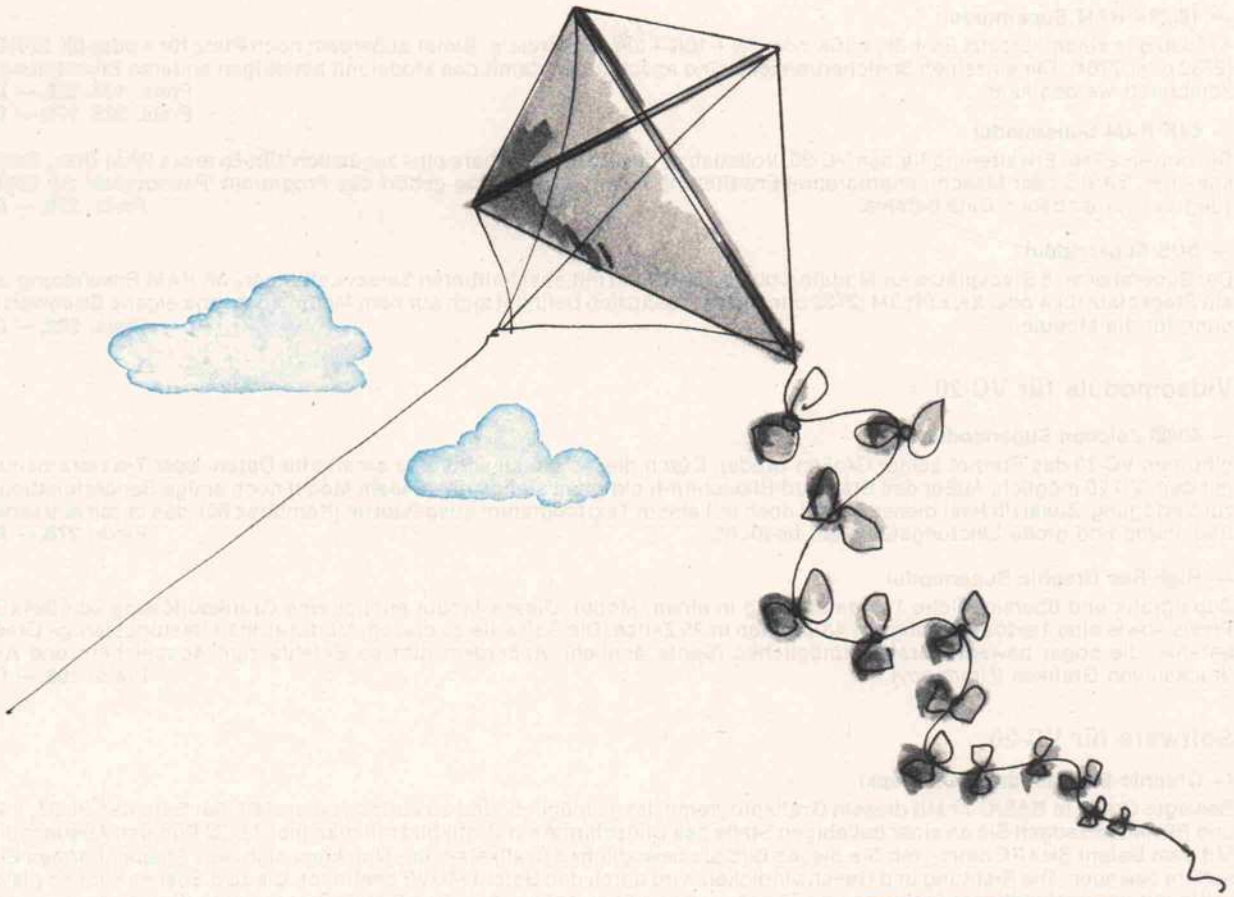
Leserbriefe

sind willkommen. Bitte schreiben Sie an die

c't-Redaktion, Postfach 27 46, 3000 Hannover 1.

Technische Anfragen beantworten wir gern, auch wenn sie sich nicht unmittelbar auf einen c't-Artikel beziehen — allerdings können wir keine Sonderentwicklungen ausführen. Wer noch vor der Veröffentlichung eine Antwort haben möchte, wird gebeten, einen frankierten, adressierten Rückumschlag beizulegen.

Aufwind durch Fortbildung — mit Christiani Lehrgängen



Christiani
Fortbildung
Technisches Lehrinstitut
Postf. 35 55150 · 7750 Konstanz
in Österreich: Ferntechnikum 6901 Bregenz
Schweiz: Lehrinstitut Onken, Kreuzlingen

Fordern Sie gleich das kostenlose Kursprogramm an, das Sie über unsere Lehrgänge informiert — u. a. auch über:

- ☐ Mikroprozessortechnik
- ☐ Mikroprozessor System 85
- ☐ BASIC mit dem VC 20
- ☐ BASIC + Mikrocomputerpraxis
- ☐ Kompakt-Kurs EDV
- ☐ Automatisierung
- ☐ Englisch / Französisch
- ☐ Elektronik-Labor
- ☐ IC-Labor
- ☐ Digital-Labor
- ☐ Elektr. Steuerungstechnik
- ☐ Videotechnik
- ☐ Elektronisches Messen
- ☐ Amateurfunk-Lizenz

Senden Sie mir kostenlos und völlig unverbindlich das neue Christiani Kursprogramm. Für den angekreuzten Lehrgang interessiere ich mich besonders.

Name, Vorname

Straße, Nr.

PLZ, Ort 85150

MicroComputerSysteme GmbH

Sonderliste Hard- und Software für VC-20 und C-64 1/84

1. Speichererweiterungen für VC-20

— 16/32K RAM Supermodul

4 Module in einem. Ersetzt 3K + 8K + 16K oder 8K + 16K + 8K(AO Bereich). Bietet außerdem noch Platz für 4 oder 8K EPROM (2732 oder 2764). Die einzelnen Speicherbereiche sind abschaltbar, damit das Modul mit beliebigen anderen Erweiterungen kombiniert werden kann.
Preis: 16K **128,— DM**
Preis: 32K **178,— DM**

— 64K RAM Supermodul

Die universellste Erweiterung für den VC-20. Vollausbau des Hauptspeichers plus zusätzlich 40K-Byte als RAM-Disk, Grafikspeicher, BASIC oder Maschinenprogramm-Erweiterung. Zum Lieferumfang gehört das Programm 'Pseudodisk' zur Ergänzung der vorhandenen Dateibefehle.
Preis: **278,— DM**

— BUS Supermodul

Der Superspacer. 5 Steckplätze für Module, voll gepuffert und mit abschaltbaren Select-Leitungen, 3K RAM Erweiterung und ein Steckplatz für 4 oder 8K EPROM (2732 oder 2764). Zusätzlich befindet sich auf dem Modul noch eine eigene Stromversorgung für die Module.
Preis: **228,— DM**

2. Videomodule für VC-20

— 40/80 Zeichen Supermodul

gibt dem VC-20 das Format seiner Großen Brüder. Durch dieses Modul wird eine ernsthafte Daten- oder Textverarbeitung mit dem VC-20 möglich. Außer den Standard-Bildschirmfunktionen stehen mit diesem Modul noch einige Sonderfunktionen zur Verfügung. Zusätzlich ist dieses Modul noch mit einem Textprogramm ausgestattet (Kombitex 80), das durch einfachste Bedienung und große Leistungsfähigkeit besticht.
Preis: **278,— DM**

— High-Res Graphic Supermodul

Supergrafik und übersichtliche Textdarstellung in einem Modul. Dieses Modul erlaubt eine Grafikauflösung von 384x256 Pixels sowie eine Textdarstellung mit 48 Zeichen in 25 Zeilen. Die Software zu diesem Modul enthält leistungsfähige Grafikbefehle, die sogar bewegte Grafik ermöglichen (Sprite ähnlich). Außerdem gibt es Befehle zum Abspeichern und Ausdrucken von Grafiken (Hardcopy).
Preis: **198,— DM**

3. Software für VC-20

— Graphic Supermodul (ROM Pack)

Bewegte Grafik in BASIC ?? Mit diesem Grafikprogramm ist es möglich. Und so einfach ist es: Mit den Befehlen PLOT, LINE und FRAME erzeugen Sie an einer beliebigen Stelle des Bildschirms ein Grafikbild mit maximal 32x32 Punkten Ausdehnung. Mit dem Befehl SHAPE definieren Sie dieses Bild als bewegliches Grafikelement. Nun kann sich dies Shape über den Bildschirm bewegen. Die Richtung und Geschwindigkeit wird durch den Befehl MOVE bestimmt. Bis zu 8 Shapes können gleichzeitig mit unterschiedlicher Richtung und Geschwindigkeit bewegt werden. Das BASIC-Programm läuft unterdessen ungehindert weiter (z. B. zur Abfrage des Joysticks). Wenn die Größe der Shapes nicht ausreicht, so besteht die Möglichkeit, auch ganze Bildschirmseiten mit 176x160 Pixels zur Darstellung von Bewegungen zu benutzen. Der Befehl SHOW kann bis zu 16 Seiten pro Sekunde anzeigen und somit ruckfreie Bewegungen erzeugen. Daß Grafikbilder ausgedruckt (Hardcopy) und auf Diskette abgespeichert werden können, sei bei soviel Leistung nur am Rande erwähnt. Graphic Toolkit belegt 8K-Byte ab $A000.
Preis: **128,— DM**

— Screen 40 plus (Cassette)

40 Zeichen und 25 Zeilen auf dem VC-20 nur per Software. Dieses Programm macht es möglich. Alle Bildschirmfunktionen bleiben erhalten (Editor, print, input). Zusätzlich stehen einige einfache Grafikbefehle zur Verfügung, mit denen beliebige Grafiken erstellt werden können. Das Programm erfordert mindestens ein 8K RAM-Modul.
Preis: **49,— DM**

— Kombitex 80 (Cassette oder Diskette)

Die Textverarbeitung für den VC-20 mit 80 Zeichenkarten. Läuft mit allen 80 Zeichenkarten, bei denen sich der Bildschirmspeicher mit PEEK auslesen läßt. Dieses Programm bietet alles, was Sie benötigen und ist dabei so einfach zu bedienen, daß Sie von jedem nach 10 Minuten Einarbeitung benutzt werden kann. Am besten läßt sich mit Kombitex 80 arbeiten, wenn Sie unser 64K RAM Supermodul und unser 40/80 Zeichen Supermodul haben, denn dann stehen Ihnen bis zu 792 Textzeilen zur Verfügung, und das Programm befindet sich im EPROM des 40/80 Zeichen Supermoduls.
Preis: **99,— DM**

— Pseudodisk Dateiprogramm (Cassette)

Floppybetrieb mit Supergeschwindigkeit, auch wenn Sie gar kein Floppydisk besitzen. Dieses Programm läuft mit unseren 64K RAM Modulen ES1 und ES2. Die Verwendung in Verbindung mit 64K Modulen anderer Hersteller ist auch möglich, wenn die Bankselectadresse in Bereich I/0 2 liegt. Pseudodisk enthält Erweiterungen für die vorhandenen Dateibefehle (OPEN, CLOSE, PRINT#, INPUT#, GET#, LOAD, SAVE), um Dateien im Hintergrund-RAM zu verwalten. Der OPEN Befehl wurde so verändert, daß auch RANDOM Dateien möglich sind, der LOAD Befehl kann auch für APPEND benutzt werden. Ab 1. 2. 84 gehört Pseudodisk zum Lieferumfang unseres 64K RAM Supermoduls.
Preis: **49,— DM**

MicroComputerSysteme GmbH

4. Z 80A Supermodul für VC-20

Dies ist eigentlich kein Erweiterungsmodul, sondern ein eigenständiger Computer, der den VC-20 als Ein-/Ausgabe Terminal benutzt. Das Modul besteht aus folgenden Funktionsgruppen:

— Z 80A CPU, Taktfrequenz 4 MHz
— 64K dynamisches RAM als Arbeitsspeicher
— 8K EPROM mit Terminalprogramm und Urlader
— DMA I/0 Interface zum VC-20

Ein Betriebssystem muß von Diskette oder Kassette geladen werden, damit das Modul seine Funktion aufnimmt. Wir bieten zwei verschiedene Betriebssysteme zu diesem Modul an. Ein CP/M 3.0 kompatibles Diskettenbetriebssystem SUPER VC-DOS und einen TRS 80 Simulator. Das Betriebssystem SUPER VC-DOS erfordert auf jeden Fall eine Diskettenstation VC-1540/41 oder unseren Floppycontroller mit mindestens einem Laufwerk. Das Betriebssystem TRS-80 Simulator ermöglicht es, TRS-80 Level II BASIC Programme ohne Änderung auf einem VC-20 laufen zu lassen. Als Massenspeicher kann dabei wahlweise eine Datasette oder eine Diskette benutzt werden. Das Z 80A Supermodul belegt die VC-20 Adressen $4000-$5FFF, $9900-$99FF und $9C00-$9FFF. Mit dem SUPER VC-DOS stehen dem Benutzer 62K-Byte Speicher zur Verfügung (2K BIOS). In Verbindung mit dem TRS-80 Simulator sind 48K-Byte Anwender-Speicher frei (16K Basicinterpreter).

Preis: **398,— DM**

SUPER VC-DOS, CP/M 3.0 kompatibel — Preis: **348,— DM**
TRS-80 Simulator, Level II kompatibel — Preis: **148,— DM**

5. Floppydisk-Controller zum Z 80A Supermodul

Floppydisk-Controller zum Anschluß von bis zu 3 Floppydisklaufwerken an das Z 80A Supermodul. Dieses Modul wird als Erweiterung an das Z 80A Modul angesteckt und kann 3 Mini-Floppy-Laufwerke steuern. Je nach verwendetem Laufwerk stehen 184K-Byte (40 Track, Single Side) bis 737K-Byte (80 Track, Double Side) pro Diskette zur Verfügung. Die Übertragungsrate beträgt 250K-Bit pro Sekunde und ist damit um den Faktor 25 schneller als beim Standardlaufwerk VC-1541. Zum Lieferumfang gehört ein Anschlußkabel für 3 Laufwerke, ein Betriebssystem-EPROM sowie eine Diskette zur Konvertierung der Betriebssysteme

Preis: **348,— DM**

Dazu können wir auch einzelne Diskettenlaufwerke oder komplette Disk-Stationen mit Gehäuse und Netzteil liefern.

Urheberrechtsnotiz: CP/M 3.0 ist ein eingetragenes Warenzeichen von Digital Research. TRS-80 u. Level II Basic sind eingetragene Warenzeichen der TANDY Corp. Z 80A ist ein eingetragenes Warenzeichen der ZILOG Inc.

6. Commodore 64 Module

— Adapter zum Anschluß von VC-20 Modulen an den C-64

Dieser Adapter ermöglicht den Betrieb folgender Module am Commodore 64: 64K RAM Supermodul — 40/80 Zeichen Supermodul — Z 80A Supermodul + Floppycontroller. Der Busadapter kann mit zwei Steckplätzen oder mit vier Steckplätzen geliefert werden. Die Version mit 4 Steckplätzen enthält noch eine eigene Stromversorgung für die Module.

Preis mit 2 Steckern: **78,— DM**
Preis mit 4 Steckern: **148,— DM**

— 80 Zeichen + Grafik Supermodul

Dieses völlig neue Modul setzt neue Dimensionen. Nicht nur 80 Zeichen pro Zeile, sondern auch noch Supergrafik mit 256000 Pixel Auflösung (640x400 Punkte). Die Software enthält viele Grafikbefehle und Befehle für Hardcopy und zum Abspeichern von Grafikbildern auf Diskette. Der Textbildschirm umfaßt 4000 Zeichen, die wahlweise auf einer oder auf zwei Bildschirmseiten dargestellt werden können.

Preis: **348,— DM**

— Screen 80 plus (Cassette oder ROM)

80 Zeichen und 25 Zeilen nur per Software. Dieses Programm ermöglicht ein übersichtliches Bildschirmformat und enthält außerdem einen komfortablen Grafik-Befehlssatz zum Erstellen beliebiger Grafiken im Format 320x200. Das Programm belegt 7K des internen RAM Bereichs. Es kann auf Cassette oder als ROM Modul geliefert werden.

Preis auf Cassette: **98,— DM**
Preis auf EPROM: **148,— DM**

7. Peripherie für VC-20 und C-64

— Datenmonitor BMC BM12

Dieser Monitor ist mit einer Bandbreite von 18 MHz das ideale Ausgabegerät für eine gestochen scharfe 80 Zeichendarstellung. Lieferbar mit grüner oder bernsteinfarbener Bildröhre.

BMC BM12 ES (grün) Preis: **368,— DM**
BMC BM12 EY (bernstein) Preis: **398,— DM**

— Matrixdrucker BMC BX-80

Der Drucker für Ihren Commodore. Bietet alle Funktionen für einen professionellen Einsatz und ist dabei sehr preisgünstig. Einige Besonderheiten: Superschriftbild durch 9x13 Matrix, Friktionswalze und Traktorführung, sehr leise, 80 Zeichen/Sek. 40-80-142 Zeichen/Zeile, Grafikauflösung 640 oder 1280 Punkte pro Zeile, Unterstreichen, Fettdruck, Subscribt u. Superscribt (Hoch- und Indexschrift), umfangreiche Tabulierfunktionen. Lieferbar mit Centronics Schnittstelle, V-24 oder VC-20/C-64 Interface.

Preis mit Centronics: **998,— DM**
Preis mit V-24/RS 232c: **1148,— DM**
Preis mit VC Interface: **1148,— DM**

Ladenverkauf Raum Bremen
Microcomputersysteme GmbH
Bremer Str. 23
2807 Achim
Tel.: 0 42 02/8 31 31

Ladenverkauf Raum Frankfurt
Königsteiner-Funk-Center
Wiesenstr. 18
6240 Königstein
Tel.: 0 61 74/2 19 53

Versandanschrift
Microcomputersysteme GmbH
Kirchweg 5
2831 Schwaförden
Tel.: 0 42 77/6 92

Hobby-tronic '84

Computer, Computer

Der Heimcomputerboom zu Weihnachten zeigte es bereits überdeutlich: Das Hobby 'Computer' verdrängt immer mehr seine altehrwürdigen Ursprünge aus dem analogen Bereich. So hat es wohl kaum jemanden verwundert, daß 'die Produktgruppe Mikrocomputer' in diesem Jahr mit fast 50% den Schwerpunkt des Ausstellerangebots auf der Dortmunder Hobby-tronic bildete.

Nach Mitteilungen des Ausstellungspressedienstes war die Zahl der Aussteller mit 137 zwar geringfügig rückläufig, die vermietete Ausstellungsfläche jedoch hat sich weiterhin vergrößert. Neben den Ausstellern beteiligten sich sechs Elektronik-Freizeitverbände an der Hobby-tronic. Sie unterrichteten im 'Action-Center' firmenneutral über verschiedene Elektronik-Bereiche wie Mikrocomputer, CB-Funk, Video- und Tonband, Kurzwellen-Hörer und Amateurfunk. Mit 69148 Besuchern wurden die Vorjahreszahlen um fast 20% überschritten. Soviel zu den Zahlen.

Wenn man in Rechnung stellt, daß die Hobby-tronic in erster Linie eine Verkaufsmesse ist, dann gab es doch erfreulich viel Neues zu sehen.

Beschränkte man sich allerdings auf die kursierenden Pressemitteilungen, so war die einzige Computersensation ein Flugsimulationsprogramm, geschrieben für den IBM PC, vorgeführt auf dem Genie 16 der Firma Trommelschläger. Mit diesem Programm konnte man mehrere Stunden lang den Flug einer Piper 181 von Chikago nach Seattle nachvollziehen. Nacht- und Tagflug und diverse Wetterverhältnisse werden vom Programm berücksichtigt.

Mindestens ebenso interessant aber ist der Genie 16 selbst, ein farbgrafikfähiger echter 16-Bit-Computer, der zudem IBM-kompatibel ist und für unter 6000 D-Mark angeboten wird.

Die Firma Räbiger Mikrocomputersysteme stellte den Tatung TPC 2000 vor, ein CP/M-fähiges Bürokomplettsystem. In der Grundversion ist er mit zwei 5,25"-Laufwerken à 360 KB ausgestattet. Schreibmaschinen-Tastatur und Monitor (12 Zoll, 22 MHz, 19,2 kHz Zeilenfrequenz) sind ebenso im Grundpreis (rund 5200 D-Mark) enthalten wie CP/M 2.2, MBASIC und ein Textverarbeitungsprogramm.

Sogar den Macintosh durfte man, wenn auch nur am Händlertag, auf dem Apple-Stand bewundern. Anschließend wurde er nur noch erlesenen Interessenten 'in Klausur' vorgeführt, Zitat: 'Wir sind schließlich Händler'.

Bei den 'kleinen' Computern wurde auch einiges geboten. So war der neue ORIC ATMOS (Testbericht in diesem Heft) zu sehen und zu hören, unter anderem in Verbindung mit dem c't-Sprachsynthesizer.

Mit dem Dragon 64 (64K RAM) will sein Hersteller jetzt den Einstieg ins Profilager wagen. Als Diskettenbetriebssystem wurde für den 64 das OS-9, eine UNIX-Ableitung, konzipiert. Für das OS-9 werden ein Spreadsheet-Kalkulationsprogramm, ein Textverarbeitungsprogramm mit Rechtschreibungsprüfung und ein Dateiverwaltungsprogramm erhältlich sein.

Und endlich ist auch der langerwartete MTX 512 von Memotech, dem wir in diesem Heft einen eigenen Beitrag gewidmet haben, eingetroffen.

Die Computer des Herstellers Spectra Video (Hongkong/New York), der SV-318 und der SV-328, waren ebenfalls vertreten. Seit der Hobby-tronic sind die Preise für die beiden sogar noch gesenkt worden. Der 318, der durch sein Steuerknüppel-Design mehr an eine Videospielkonsole denn an einen Computer erinnert, kostet derzeit rund 800 D-Mark, der bei entsprechender Aufrüstung CP/M-fähige 328 etwa 1100. Beiden gemeinsam ist, daß sie für Software nach dem MSX-Standard (MicroSoft eXtended) ausgelegt sind.

Leider waren aber auch einige langerwartete Neuheiten nicht zu sehen: die sagenumwobenen Micro-Drives der Firma Sinclair ebensowenig wie der QL.

Fast aufdringlich wirkte die Masse von Apple-Nachbauten. Jedesmal, wenn man gehofft hatte, wieder einen unbekannten Leckerbissen womöglich gar aus deutscher Fertigung, entdeckt zu haben, befand sich das altbekannte Früchtchen im Innern. Als Portable im Osborne-Look, als Büro-Computer 'getarnt', die äußerliche Vielfalt war beeindruckend, fast könnte man von einer Designstudie sprechen. Angenehm daran ist allerdings der immer weiter sinkende Preis.

Der Bereich der 'Computerbestandteile' bot ebenfalls manches Erwähnenswerte. So zum Beispiel neue intelligente Tastaturen: Die 'Operator' von der Firma AFC (430 D-Mark) und die KBE 4.X von SH-electronic (549 D-Mark), die trotz ihrer vielfältigen Programmiermöglichkeiten preislich in der Reichweite des Hobbyisten liegen.

Nicht zu vergessen die ungeheure Menge von preiswerten Monitoren (zwischen 200 bis 400 D-Mark), deren Bezugsquellen wir hier unmöglich alle aufführen können. Bemerkenswert ist allerdings der hochauflösende RGB-Farbmonitor CT 900 HR 14, den die Firma Trommelschläger vertreibt. Für rund 2300 D-Mark bietet dieser Monitor mit einer Auflösung von 720x290 Pixeln ein erstaunlich flimmerfreies und augenfreundliches Bild.

Interessante Randerscheinungen: Das Höchstalter für die Teilnehmer am Wettbewerb 'Jugend programmiert' war auf 16 Jahre herabgesetzt worden. Tatsächlich scheint das Einstiegsalter der Computerfreaks immer weiter zu sinken.

Und dann gab es natürlich jede Menge Bauteile-Händler, die besonders für die Computerfreaks immens wichtig waren, da sich hier eine glänzende Möglichkeit eröffnete, seltene Bauteile, wenn auch gelegentlich zu Schwarzmarktpreisen (7406 für 8 Mark), zu beschaffen.

Man traf selbstverständlich auch auf weniger Spektakuläres. An einigen Ständen drängte sich einem der Eindruck auf, hier sei nur für die Messe eine Tonne Computerschrott aufgekauft worden, der dann pfundweise verhökert wurde. Vielfach lagen auch Schrott und Fortschritt einträchtig nebeneinander, zum Beispiel ausgeschlachtete Trafos neben Schaltnetzteilen aus laufender Fertigung.

Und wie groß das Interesse an all dem war, bekundeten nicht nur die Verkaufszahlen. Um es ganz unverblümt zu sagen: Alles, was nicht massiv verschraubt war oder nicht gerade einen Gabelstapler zum Transport benötigte, wurde geklaut. Aber das ist nun ganz und gar nichts Neues.

Anschriften zu namentlich aufgeführten Produkten:

Genie 16, Farbmonitor CT 900 HR 14:

TCS Computer GmbH
Kölnstraße 2—4
5205 St. Augustin 2

Tatung TPC 2000:

P. Räbiger
Microcomputer-Systeme
Veldener Str. 65, 5160 Düren

ORIC Atmos:

MSE electronic
Bonner Str. 103
4000 Düsseldorf 13

UTAW electronic
Hagenstr. 31
3000 Hannover 1

Dragon 64:

Norcom Vertriebs-GmbH
Postfach 33 28, 8500 Nürnberg

SV-318/328:
ptm Elektronik GmbH
Am Stimmbeck 2
2730 Heeslingen

OPERATOR:

AFC Computer GmbH
Salmstr. 20, 5000 Köln 91

KBE 4.x:

SH-electronic
Hauptstr. 204, 5488 Adenau

MTX 512:

profisoft
Sutthauser Str. 50—52
4500 Osnabrück

Hazeltine Esprit

EIN NEWCOMER MIT 13 JAHREN ERFAHRUNG

Aus der Hazeltine Terminal Division entstand Esprit Systems Inc. So kann dieses Unternehmen, obwohl ein neuer Name auf dem Elektronikmarkt, doch auf die Erfahrung eines bekannten Terminal-Herstellers verweisen. Aufbauend auf einer bewährten Produktpalette wird Esprit Neuentwicklungen und einen optimierten Kundendienst anbieten. Sie haben also die Auswahl unter einem breiten Spektrum leistungsfähiger Geräte: IBM-kompatible Terminals ebenso wie flexible OEM-Systeme mit farbigen oder monochronen Bildschirmen. Egal, für welche Version Sie sich letztendlich entscheiden: Sie bekommen ein Produkt, das sich durch ein bestmögliches Preis-/Leistungsverhältnis auszeichnet. So z. B. unser neues, DEC-kompatibles Terminal Executive 10/102, das eine neue Dimension ergonomischen Designs in seine Preiskategorie einbringt.

Autorisierter Distributor für die Bundesrepublik Deutschland

SYNELEC
DATENSYSTEME GmbH

Lindwurmstr. 117 · 8000 München 2
Telefon 0 89/7 25 30 81 · Telex 5 212 289 syn d
Geschäftsstelle Düsseldorf: Telefon 02 11/35 02 36
Geschäftsstelle Frankfurt: Telefon 0 60 81/4 21 55

Antistatic auf dem Tisch

Die Tischauflage Touch-Stat soll die durch statische Elektrizität verursachte Gefährdung von Rechnersystemen beseitigen. Touch-Stat ist eine Tischmatte, die durch eine Erdungsleitung mit der Erdungsklemme einer Schuko-Steckdose verbunden wird. Die Systemkomponenten werden dann auf die Matte gestellt.

Informationen: Dennison International Company, Matthias-Claudius-Str. 9, 4006 Erkrath 1

CAD mit 16-Bit

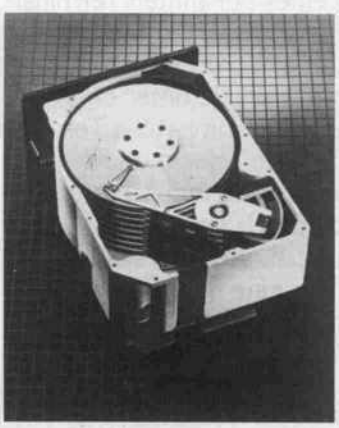
Die Firma Energotec bietet ein kostengünstiges CAD-System für die Erstellung von elektrotechnischen Schalt- und Steuerungsplänen an. Die CAD-Software läuft auf zwei Computern, die den 8086-Prozessor verwenden: Duet-16 von Panafacom und Altos 586.

Informationen: Energotec Datentechnik GmbH, Grafenberger Allee 130a, 4000 Düsseldorf 1

5¼"-Festplatte mit 140 MByte

Die 5¼"-Festplattenlaufwerke der Serie XT 1000 sind mit Kapazitäten von 65, 105 und 140 MByte erhältlich. Alle drei Versionen haben eine mittlere Zugriffszeit von 30 ms. Damit steht erstmalig ein Festplattenlaufwerk in 5¼"-Abmessungen mit mehr als 100 MByte zur Verfügung.

Informationen: Metrologie GmbH, Hansa Str. 15, 8000 München 21



32-Bit-CPU von Zilog

Zilog stellt eine neue 32-Bit-CPU vor, die bis zu fünf Millionen Befehle pro Sekunde ausführen kann. Der Prozessor Z80.000 verwendet intern und extern 32-Bit-Datenwege und ist voll kompatibel zu der Z8000-CPU. Der neue Mikroprozessor kann vier Giga-Byte Speicher direkt adressieren und arbeitet mit Taktfrequenzen von 10 MHz bis 20 MHz. Eine hohe Daten-Durchsatzgeschwindigkeit gewährleistet unter anderen ein integrierter 256-Byte-Cache-Speicher, der sowohl Daten als auch Programm-Code enthalten kann. Muster der Z80.000-CPU stehen laut Zilog im zweiten bis dritten Quartal 1984 in kleinen Stückzahlen zur Verfügung.

Informationen: Zilog GmbH, Eschenstraße 8, 8028 Taufkirchen

VT100 Emulation mit Sirius

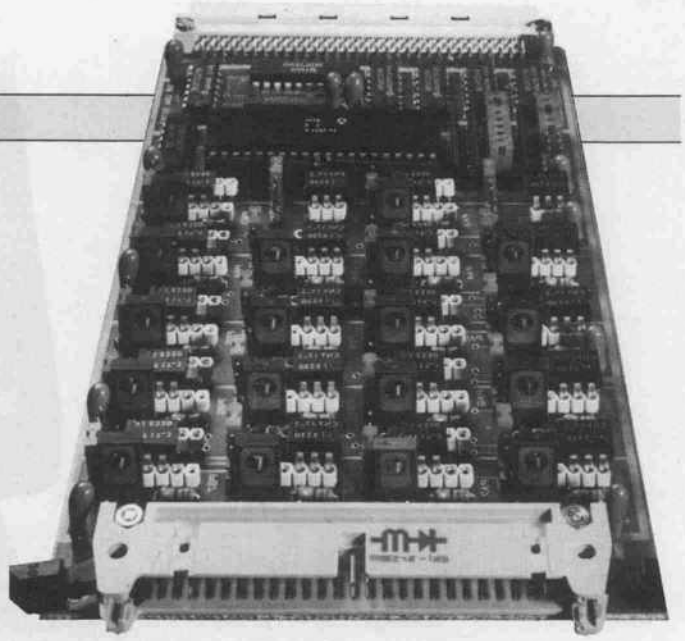
VT100-Emulation von UTS erlaubt es, den Microcomputer Sirius 1 an Mini- und Großcomputer von Digital Equipment anzuschließen. Damit hat der Anwender die Möglichkeit, auf die zentrale Datenbank des Großrechners zuzugreifen.

Informationen: UTS Ltd. KG, Orber Straße 24, 6000 Frankfurt 61

'Mega Super' mit 512 K RAM

Der Einplatinencomputer 'Mega Super' arbeitet mit einer Z80-CPU (5 MHz Systemtakt). Sein Arbeitsspeicher kann auf maximal 512 KByte ausgebaut werden und ist entweder als RAM-Disk oder als 'banked RAM' zu betreiben. Neben sechs seriellen und zwei parallelen Schnittstellen steht eine PRIAM/SASI-Harddisk-Schnittstelle, ein Z80 CTC, ein Z80-DMA-Controller und ein AMD 9511 'Math-Processor' zur Verfügung. Der Controller WD 2797 unterstützt maximal drei Floppy-Laufwerke. Als Betriebssystem dient CP/M oder MP/M.

Informationen: Statistics + Software, Bureau Wolfram Fassbender, P.O.B. 132, 1000 Berlin 42



20 mal I/0 auf EUROBUS

Zwanzig programmierbare I/0-Kanäle stellt die Peripheriekarte 10235-POT bereit. Jeder Kanal kann als Ein- oder Ausgang beschaltet werden. Die eigentliche Datenein- und Ausgabe geschieht, galvanisch getrennt, durch Optokoppler. Ist ein Kanal als Ausgang definiert, so ist dem Optokoppler ein Transistortreiber nachgeschaltet, der die direkte Lastbeschaltung bis 30 V/0,5 A erlaubt. Die Karte entspricht in ihrem Pinout dem EUROBUS-Standard.

Informationen: EKF Elektronik Meßtechnik GmbH, Weidekampstr. 1A, 4700 Hamm 1

Zweifarbendruck mit Laser

Das Laserdrucksystem der Firma Siemens erlaubt den zweifarbigen Druck (schwarz und blau) bei einer Druckgeschwindigkeit von 100 DIN-A 4-Seiten pro Minute. Außerdem kann es

in einem Drucklauf die Vor- und Rückseite einer Papierbahn bedrucken. Das Drucksystem besteht aus zwei zusammenarbeitenden Einzeldruckwerken.

Informationen: Siemens AG Info-Service, Postfach 156, 8510 Fürth

Einkaufen in USA

Die Firma Orbit bietet ihren Kunden den direkten Zugriff auf das amerikanische Elektronik-Angebot. Die Firma kauft im Kundenauftrag Ware zum US-Inlandspreis. Da US-Waren in Europa zwischen 20 und 50% teurer sind als im Ursprungsland, stellt das Angebot eine Möglichkeit dar, schnell und günstig an benötigtes Material zu gelangen.

Informationen: Orbit GmbH, Postfach 28, 8082 Grafrath b. München

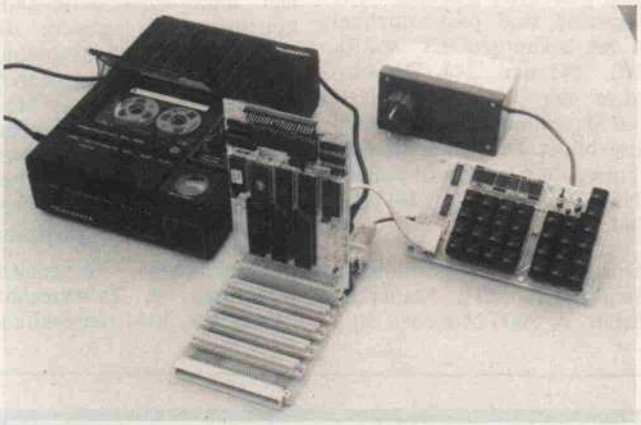
LPRINT III für ZX Spectrum

Das Interface LPRINT III bietet allen ZX-Spectrum-Anwendern die Möglichkeit, Drucker mit paralleler oder serieller Schnittstelle an ihrem Rechner zu betreiben. Die komplette Treibersoftware für alle gängigen Drucker ist bei LPRINT III bereits in einem EPROM gespeichert. Mit verschiedenen 'POKE'-Befehlen kann das Interface optimal an den verwendeten Drucker angepaßt werden.

Informationen: Microcomputer Laden, Kantstr. 70, 1000 Berlin 12

COBOLD

IHR Lern- und Proficomputer auf drei Platinen!
Der ideale Einstieg in die Microprozessortechnik



COBOLD — ein Computer mit zauberhaften Qualitäten dank eines neuen, raffinierten Hardware-Konzepts und eines sagenhaft komfortablen Betriebssystems. Auf drei Platinen.
— ein Maschinensprache-Computer auf Basis 6502/65C02, der auch Textverarbeitung, BASIC und FORTH kann.
— der sinnvollste Einstieg in die Microprozessortechnik.
— der Computer für alle — auch Ihre — Problemstellungen.
— beschrieben mit Bauanleitung in ELRAD 3, 4 + 5/83.

Lernen auch Sie zaubern wie ein Cobold — steigen Sie ein in die Microprozessortechnik mit dem neuen elrad-COBOLD-System! Fordern Sie Prospekte an!

Die Komplett-Ausstattungen:

GRUNDVERSION: (CIM 65-Prozessorkarte, Basis- und TD-Platine) mit CPU 6502. RIOT 6532, 2 K RAM, Monitor-EPROM. Basisplatine bestückt mit 1 Federleiste.

Bausatz	**DM 298,—**
Bausatz mit fertiger CPU-Karte	**DM 389,—**
Fertig aufgebautes System	**DM 449,—**

ERWEITERTE VERSION (Grundversion mit 4 K RAM, 3x RIOT 6532, Basisplatine mit 5 Federleisten).

Bausatz	**DM 398,—**
Bausatz mit fertiger CPU-Karte	**DM 498,—**
Fertig aufgebautes System	**DM 549,—**

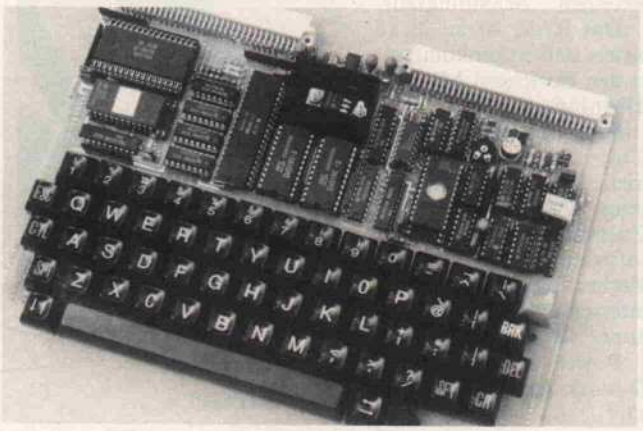
NETZTEIL für den COBOLD im Steckergehäuse **DM 49,—** (Bausatz) bzw. **DM 86,—** (fertig) (auch für c't-Terminal geeignet).

DAS HANDBUCH für den COBOLD: „6502/65C02 Maschinensprache" von C. Persson **DM 48,—.**

Intelligentes Terminal
mit professionellen Attributen:

c't-Terminal

DER Terminal-Computer auf Doppel-Euro-Karte
mit oder ohne integrierter Tastatur!

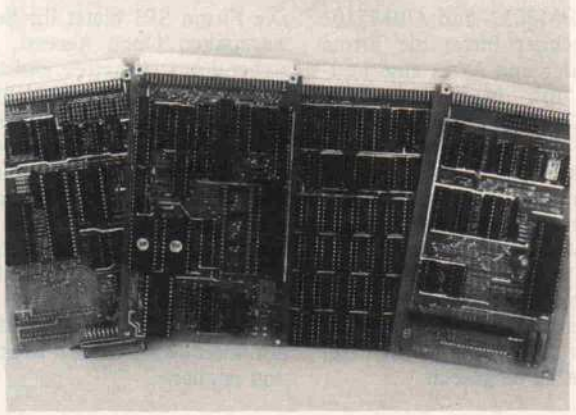


— beschrieben in c't Nr. 12/83 und 1/84
— 6511-Singlechipcomputer mit 6545-Videocontroller
— 4 KB-Bildwiederholspeicher (scrollbar)
— Bildformat 80 x 25 oder 64 x 20 (per Software umschaltbar)
— Zeichenmatrix 8 x 11 (bei 80 x 25) oder 8 x 13 (bei 64 x 20)
— max. 8 Zeichensätze (inkl. Blockgrafik)
— Invers-, Blink-Modus, Breitschrift, halbe Helligkeit
— serielles Interface (V 24- oder TTL-Pegel)
— integrierte Centronics-Schnittstelle
— integrierte Spannungsregelung und -wandlung für V24
— 8-bit-parallel (ASCII) oder 8 x 9 Tastenmatrix Tastaturanschluß
— PREISE: **Version A** (ohne Tastatur)
Bausatz **DM 449,—; DM 549,—** Fertigkarte
Platinenmaße 233 x 85 mm
Version B (mit integrierter Tastatur)
Bausatz **498,—; DM 639,—** Fertigkarte
Platinenmaße 233 x 160 mm

Prospektmaterial auf Anforderung!

Endlich: c't-86

Das 1. echte 16-bit-Microcomputer-System
der Welt (auch) zum Selbstbau!



Ein Vier-Karten-System, basierend auf dem für 16 bit erweiterten ECB-Bus:
— echte 16-bit-Rechenleistung
— kein neuer, sondern ein weitverbreiteter Bus
— dadurch bereits existierende ECB-Peripherie-Karten einsetzbar
— Betriebssysteme CP/M-86 und MS-DOS II
— **Komplettpreis unter DM 1 700,— (Bausatz)**
— vorgestellt in Heft 1, 2 + 3/84 von c't — dem neuen Magazin für Computertechnik

Die vier Karten:

Platine 1: **CPU-KARTE mit 8086,** optional 8087 Arithmetik-Prozessor, 8259 Interrupt-Controller, 8 KB Monitorprogramm mit CP/M-86-Urlader.
Komplett-Bausatz ..... **DM 349,—; DM 449,—** Fertigkarte

Platine 2: **I/O-KARTE** mit V-24-Interface für Terminal-Anschluß, Centronics-Schnittstelle, Kassettenrekorder-Interface und Timer.
Komplett-Bausatz ..... **DM 249,—; DM 349,—** Fertigkarte

Platine 3: **FLOPPY-CONTROLLER-KARTE** zum Anschluß bis zu 4 Laufwerken 5¼ oder 8 Zoll (auch gemischt) mit dem neuen Controller-IC WD 2797.
Komplett-Bausatz ..... **DM 498,—; DM 598,—** Fertigkarte

Platine 4: **256-KB-RAM-KARTE** mit 128 oder 256 KB dyn. RAM (max. 3 Karten einsetzbar ≙ 768 KB RAM!).
Komplett-Bausatz **DM 598,—** (128 KB) bzw. **DM 899,—** (256 KB), **DM 698,—** bzw. **DM 999,—** Fertigkarte

Geplante Ergänzungen: CPU-Karte mit 68000, SASI-Interface, Vollgrafikkarte, Z80 Subprozessor-Karte.

Bus-Karte mit 10 Steckplätzen — fertig — ..... **DM 169,—**
Leerplatinen, Floppy-Laufwerke, Netzteile und Gehäuse ..... **auf Anfrage**

Komplettsystem, anschlußfähig im Gehäuse, mit Laufwerken ab DM 3985,—

Fordern Sie Prospekte an!

CEPAC-65 Version A DM 69,— (Bausatz) · Version B DM 89,— (Bausatz)

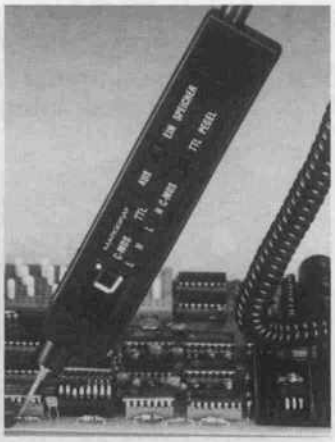
Frölje Elektronik oHG

Gaststraße 10 — 2900 Oldenburg — Telefon (04 41) 1 58 53 — 24 Std. Bestellannahme

CMOS-RAM mit Batterien

Das CMOS-RAM MK48Z02 ermöglicht Datenpufferung ohne externen Schaltungsaufwand. Das RAM ist zu 2Kx8 organisiert und ist pinkompatibel zu den Speichern MK4802 und HM6116. Die zur Pufferung nötige Spannung bezieht der Chip aus zwei Lithium-Batterien, die im Gehäuse des Bausteins untergebracht sind. Der Zustand der Batterien ist per Software prüfbar. Ein automatischer Schreibschutz tritt bei Unterschreiten der Betriebsspannung in Funktion. Die 'Zero Power RAMs' Kosten 150 D-Mark (ohne Mehrwertsteuer).

Informationen: raffel & co. electronics GmbH, Lochnerstraße 1, 4030 Ratingen 1



Universeller Logiktester

Der Logiktester LGT erlaubt es, die logischen Pegel an TTL- und CMOS-Schaltkreisen zu überprüfen. Der Zustand der IC-Ausgänge wird mit einem 7-Segment-Display angezeigt. Der Teststift kann mit Versorgungsspannungen von 4 V bis 16 V betrieben werden und ist mit einem Verpolungsschutz sowie einen Überspannungsschutz bis 42 V ausgestattet. Das Gerät verfügt über einen einschaltbaren Speicher für Impulsmessungen. Die Schaltschwellen für 'High' oder 'Low' sind zur Anpassung an die Betriebsspannung einstellbar.

Informationen: Marggraf GmbH, Bonholzstr. 24, 7035 Waldenbuch

Sichere Buffer

Speziell für den Einsatz in störempfindlichen Systemen sind die neuen Buffer von Monolithic Memories bestimmt. Die Bausteine sind pinkompatibel zu den bekannten ICs 74S210, '240, '241 und '244. Die Eingänge der 'sicheren Buffer' sind als Schmitt-Trigger-Stufen ausgebildet. Die neuen Bauteile tragen die Bezeichnungen SN54/74S310, '340, '341 und '344.

Informationen: Monolithic Memories GmbH, Mauerkircherstr. 4, 8000 München 80

Netzwerk für Ausbildung

Das Netzwerksystem 'Smallnet' ist für den Einsatz in Schulen und Ausbildungsstätten konzipiert. Es erlaubt dem Ausbilder, die Schirminhalte von 32 Mikrocomputer-Arbeitsplätzen auf seinem Monitor darzustellen. Die wesentlichen Vorteile des 'mini-Netzwerks' sind ein niedriger Preis und die Gewißheit, daß 'Systemabstürze' unmöglich sind.

Informationen: Siebeneicher Elektronik & Datentechnik, Postfach 9, 3044 Neuenkirchen

Spooler mit 8 bis 64 KByte

Über einen internen Speicher von bis zu 64 KByte verfügen die Spooler der Serie CTX. Die mit seriellen oder parallelen Interface lieferbaren Geräte sind mit zwei Tasten ausgestattet, die das Löschen des Speichers und das zwischenzeitliche Stoppen des Druckvorgangs ermöglichen.

Informationen:
Leunig GmbH Mikroelektronik
Im Wolfsgarten 10
5206 Neunkirchen-Seelscheid 1

Tastatur programmierbar

Mit beliebig langen Codes frei programmierbar ist die Tastatur 'KeyStar'. Unprogrammiert ist die Tastatur an das Textverarbeitungsprogramm WordStar angepaßt. Zum Anschluß an den Computer dienen eine serielle TTL- oder RS232-Schnittstelle und eine parallele Schnittstelle. Der Preis der Tastatur beträgt 895 D-Mark zuzüglich Mehrwertsteuer.

Informationen: oettle + reichler, A.-Stifter-Str. 40, 8902 Neusäß

Festplatte für Commodore

Für CBM8032 und CBM710/720-Rechner bietet die Firma Schneider und Koch eine Festplatte mit Multi-User- und Back-Up-System an. Die Festplatte hat eine Kapazität von 21 MByte (formatiert). Es steht ein intelligentes Betriebsprogramm dafür zur Verfügung. Das Multi-User-System kann bis zu 16 CBM-Rechner gleichzeitig bedienen. Bei dem angeschlossenen Back-Up-System werden ¼-Zoll-Cartridge-Laufwerke eingesetzt.

Informationen: Schneider & Koch & Co. Datensysteme GmbH, Postfach 58 44, 7500 Karlsruhe

'Open Access' mit Schulung

Die Firma SPI bietet ihr Softwarepaket 'Open Access', bestehend aus Disketten und Dokumentation, für DM 2500 plus Mehrwertsteuer an. Außerdem wird 'Open Access' zusammen mit einer 8stündigen Schulung, Telefonsupport und einer Softwaregarantie für ein halbes Jahr zum Netto-Preis von DM 3000 angeboten. Die Auslieferung erfolgt für die IBM-PC/XT-Version, lauffähig unter MS-DOS, in deutsch und englisch.

Informationen: Software Products International GmbH, Amsinckstraße 44, 2000 Hamburg 1

TRS 80 M 1/Video-Genie/TRS 80 M 1/Video-Genie/TRS 80 M 1/Video-Genie

Endlich wieder verfügbar: RS232 (V.24)!

Diese Einheit besteht aus zwei voneinander unabhängigen Schnittstellen. Die Einstellung der Baudrate erfolgt softwaremäßig.
Folgende Baudraten sind vorgesehen:
75, 150, 300 (Modem), 600, 1200, 2400, 4800, 9600, 19200
Die Baudrateneinstellung kann natürlich für beide Kanäle unterschiedlich eingestellt werden, wobei auch „krumme" Übertragungsgeschwindigkeiten möglich sind.
Alle nötigen Parameter, die zum Betrieb der Schnittstellen erforderlich sind (STOPBITS, PARITY, CTS, RTS, 5/6/7/8 BIT), können softwaremäßig gewählt werden.
Des weiteren stehen Ihnen noch zwei CTC-Kanäle zur freien Verwendung zur Verfügung.
Der Einbau der Karte (ca. 6,5 x 13 cm) in Ihr Gerät ist gemäß der mitgelieferten Einbauanleitung einfach durchzuführen.
Die Schnittstellenausgänge bestehen aus zwei normgerechten D-Steckbuchsen (25polig).

Nun zur mitgelieferten Software:

Es werden folgende Parameter auf Disk oder Kassette mitgeliefert:
MULTICOM ..... = vielseitig einsetzbares Softwarepaket mit Terminal-Modus und der Möglichkeit, Dateien in Hex- und in ASCII-Format auszutauschen
SPRCTS U. SPRX ..... = Druckertreiber zum Ansteuern eines seriellen Druckers über die V.24-Schnittstelle

Preis für fertig aufgebaute Platine (geprüft) inkl. Software (RB V.24 F) ..... DM **299,—**

Preis für kompl. Bausatz inkl. Schaltbild, Bestückungsplan und Software (RP V.24 B) ..... DM **249,—**

Grafik HRG1B
Mod. 1 + Video-Genie

Die von uns entwickelte HRG1B ist eine Weiterentwicklung der HRG1A, die eine Grafikauflösung von 384 × 192 Bildpunkten erlaubt. Sie kann von Ihnen selbst oder auch von uns auf- bzw. eingebaut werden. Ein- und Aufbau siehe links (EXP1). Das Einbauboard besitzt einen eigenen Speicher von 12 KByte, so daß Ihr RAM-Speicher nur von einem kleinen Treiberprogramm belegt wird.

Bei der HRG1B besteht die Möglichkeit, Ihre ASCII- + Grafik-Darstellung mit der hochauflösenden zu mischen. (Bild links.)

Sie können auch die Darstellung der HRG1B auf dem Bildschirm unterdrücken, während z. B. Ihr Basicprogramm eine Grafik erstellt. Das Treiberprogramm zur Verwaltung der HRG1B ist im Grundpreis enthalten (Kassette/Diskette). Disk + DM 8.–

Fertig ..... DM **379,—**
Bausatz ..... **Auf Anfrage**
Platine ..... DM **100,—**

Expander EXP1

Die von uns entwickelte Expanderplatine EXP1 beinhaltet folgendes:

1. Ein Floppyinterface für maximal 4 Laufwerke. Es werden sowohl ein- als auch doppelseitige Laufwerke unterstützt.
2. Eine Centronics-Parallelschnittstelle zur Ansteuerung eines Druckers, der sowohl beim TRS-80 als auch beim Video-Genie arbeitet. Sie können also druckeransteuernde Software vom Video-Genie ohne eine Änderung auf Ihrem TRS-80 laufenlassen oder umgekehrt.
3. 25 Millisekunden Interrupt zur Ansteuerung der Echtzeituhr.

EXP1 kann direkt im Tastaturgehäuse untergebracht werden.
EXP1 ist voll funktionskompatibel zu den Standard-Expandern (ohne RAMs).
Die Platine kann von Ihnen selbst oder auch von uns auf- bzw. eingebaut werden.
Der Selbstaufbau ist einfach und problemlos durchzuführen. (Durchkontaktierte Platine mit Lötstopplack inkl. Anleitung und Bestückungsplan und allen Bauteilen.)
Der Selbsteinbau besteht aus dem einfachen Anlöten der Anschlußdrähte nach Plan.
Größe der Platine nur 150 × 100 mm.
Platine aufgebaut und getestet DM 449.–

Bausatz EXP1 für nur DM **339.–**

Alle hier angebotenen Produkte sind ab Lager lieferbar und geeignet für den Einbau in TRS-80 Mod. 1, EG3003/8 und Video-Genie 1 + 2. Die Preise verstehen sich inkl. MwSt. und exkl. Versandkosten.
Günstige Händlerkonditionen. Noch Auslandsvertretung zu vergeben.
TRS-80 ist ein Warenz. der Tandy Corp.

Double-Density-Controller DBL1
erhöht die Speicherkapazität Ihrer Laufwerke um das 1,8fache inkl. Dataseparator.
DBL1 fertig DM 275.–

im Bausatz für nur DM **189.–**

MEM48
RAM-Erweiterung auf 48 KByte im Tastaturgehäuse.

Einbausatz für nur DM **178.–**

Vertretung Niederlande:
Carel Vedder Electronica
Bosstraat 102,
3971 XH Driebergen
Telefon (0 34 38) 2 07 94

RB Elektronik-Vertrieb GmbH

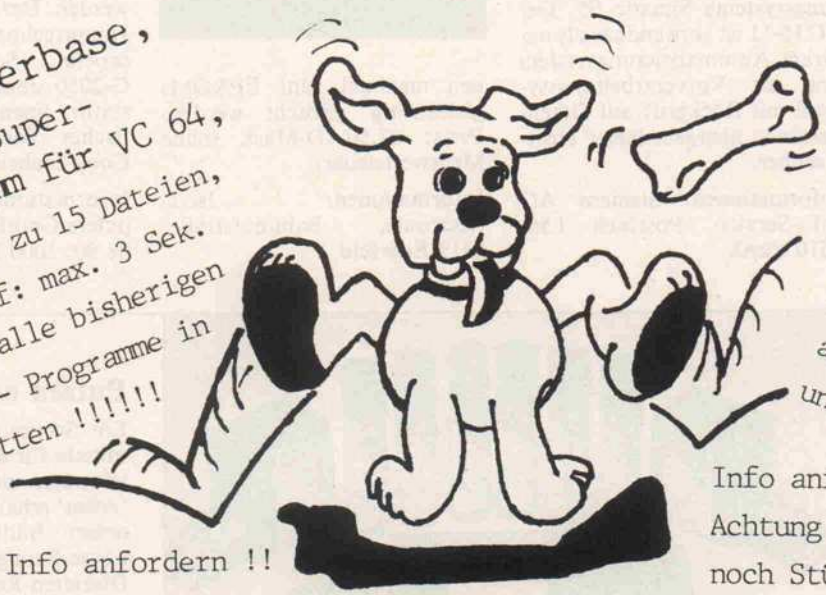
Bouraueler Straße 13, 5208 Eitorf, Telefon (0 22 43) 56 63, PF.113

Der heiße Draht
(0 22 43) 56 63

HURRA !!!! Software der Spitzenklasse

für VC20 und VC64 Spectrum , Atari, ORIC-Atmos

Und Superbase,
das Super-
Programm für VC 64,
mit bis zu 15 Dateien,
Zugriff: max. 3 Sek.
Es stellt alle bisherigen
DataBase - Programme in
den Schatten !!!!!!



Separates Info anfordern !!

Brandneu !
Der ROM-SWITCH zu
Ihrem ORIC-1 !!
Er macht aus Ihrem
ORIC einen Atmos,
und er bleibt
trotzdem ein ORIC-1 !!!
Natürlich führen wir
auch den ORIC-Atmos
und die ORIC-Micro-Drive !

Info anfordern!
Achtung ! Wir suchen
noch Stützpunkthändler !!

Unit 301 16 Brune Street
London E1 7NJ Tel.: 01-377 8034
Tlx: 896616 Sendit G

SoftShop International

MSE electronic

4000 Düsseldorf 13
Bonner Straße 103
Telefon (02 11) 79 22 62
Telex 8 582 943

IBMs PC billiger

IBM hat die Preise für Personal Computer gesenkt. Die Preissenkung beträgt je nach Ausstattung beim IBM PC circa 20 Prozent, bei dem IBM PC XT circa 10 Prozent.

Informationen: IBM Deutschland GmbH, Postfach 80 08 80, 7000 Stuttgart 80

Hilfe im EDV-Bereich

Bei der Suche nach Problemlösungen im EDV-Bereich bietet die Firma M + C Hilfe.an. Gibt der Kunde ein Stichwort an, wird unter 2000 Buchtiteln nach geeigneter Literatur gesucht.

Informationen: M + C Micro Computer GmbH, Karlstr. 17D 4018 Langenfeld

Betriebsspannung überwacht

Die ICs der Serie TL77xxA überwachen die Versorgungsspannung in Mikrocomputer-Systemen und lösen bei Unterschreitung eines bestimmten Wertes ein Reset- oder Interruptsignal aus. Das IC löst ebenfalls einen Resetimpuls aus, wenn die Betriebsspannung eingeschaltet wird. Die Bausteine sind mit festen Schwellspannungen von 4,5 V, 7,6 V, 10,8 V, 13,5 V, sowie mit einer extern einstellbaren Schaltschwelle von 2,5 V bis 18 V lieferbar.

Informationen: Texas Instruments Deutschland GmbH, Haggertystraße 1, 8050 Freising

Kurse für Mikroprozessortechnik

In der Zeit vom 24. bis 25. Mai und vom 15. bis 16. Juni veranstaltet die Firma Thaler & Co. Seminare mit den Lehrcomputern CT-65 und MPS-65. Bei den in Krefeld stattfindenden Kursen steht jedem Teilnehmer ein Computer zur Verfügung. Ziel des Lehrgangs ist, Kenntnisse über die grundlegende Arbeitsweise von Rechnern und deren Anwendung in Steuerungen und Meßwerterfassungssystemen zu vermitteln.

Informationen: THALER & Co., Mikroprozessortechnik GmbH, Magdeburger Str. 81, 4150 Krefeld

Daten und Adressen generieren

Der Adreß- und Datengenerator ADSIM 1608 soll die Fehlersuche an mikroprozessorgesteuerten Systemen erleichtern. Das Gerät stellt 16 beziehungsweise 20 Adreß- und 8 Datenleitungen bereit. Die Bitmuster an diesen Leitungen sind mit Codiertasten einstellbar.

Informationen: Labor für Elektronik u.v.G., Postfach 90 06 73, 8000 München 90

Siemens erweitert den PC 16-11

Die Firma Siemens hat ihren Personal Computer erweitert. Er kann nun gekoppelt werden an die Siemens-Rechnersysteme MMC216, 300, 6000, 7000, IBM-Rechner mit 3276-Emulation sowie an die speicherprogrammierbaren Automatisierungssysteme Simatic S5. Der PC 16-11 ist verwendbar als autarkes Automatisierungssystem und als Vorverarbeitungssystem mit Rückgriff auf Datenbestände übergeordneter Host-Rechner.

Informationen: Siemens AG Info-Service, Postfach 156, 8510 Fürth

Kodieren im Miniaturformat

Direkt von Hand und ohne Zuhilfenahme eines Schraubenziehers können die Miniaturkodierschalter von KEL geschaltet werden. Ein Drehschalter mit Sichtfenster ist auf dem eigentlichen Schaltergehäuse montiert. Die Schalter der Serie KDS 10 und KDS 12 zeichen sich durch eine kleine Bauform aus. Sie sind lieferbar in dezimaler oder hexadezimaler Kodierung sowie mit Implement- oder Komplement-Schaltung.

Informationen: Spezial-Electronic KG, Postfach 13 08, 3062 Bückeburg 1

EPROMs löschen

Isert-elektronik bietet ein EPROM-Löschgerät mit einer 12 V/4 W UV-Lampe an. Das Gerät ist mit einem elektronischen Zeitschalter ausgerüstet und in einem Aluminiumgehäuse untergebracht. Es kön-

nen maximal fünf EPROMs gleichzeitig gelöscht werden. Preis: 87,50 D-Mark (ohne Mehrwertsteuer).

Informationen: Isert-electronic, Bahnhofstraße, 6419 Eiterfeld 1

Magnetisch konstant

Die magnetischen Spannungskonstanthalter der Baureihe P unterdrücken Störfaktoren, die in jedem Netz auftreten. Die

magnetischen Konstanten stabilisieren die Netzspannung, absorbieren Störspitzen und überbrücken kurze Unterbrechungen. Die Geräte sind anschlußfertig für Leistungen von 250 VA bis 3000 VA lieferbar.

Informationen: TWK-Elektronik GmbH, Postfach 80 40, 4000 Düsseldorf 1

Interaktiver Grafik-Controller

Der Grafik-Controller G-2050 von Genisco bietet eine sichtbare Auflösung von 1024x792 Bildpunkten. Auf dem Bildschirm können 16 von 4096 Farben gleichzeitig dargestellt werden. Der Anschluß an einen Hauptrechner erfolgt über eine serielle Schnittstelle. Der G-2050 steuert intern eine Tastatur, einen Joystick, ein grafisches Tablett sowie Hard-Copy-Einheiten.

Informationen: Genisco Computers GmbH, Stolberger Straße 90, 5000 Köln 41

Putzen und pflegen

Ein Sortiment an Reinigungsmitteln für die Computerpflege ist unter dem Markennamen 'Astat' erhältlich. Dazu zählen, neben Bildschirm- und Gehäuse-Reinigungstüchern, ein Disketten-Reinigungsset und ein Antistatic-Spray. Viele Produkte des Sortiments werden vordosiert verpackt geliefert.

Informationen: HiFi-Plus Phonozubehör u. Handelsgesellschaft mbH, Postfach 28, 3548 Arolsen

SYSTECH GMBH

OLYMPIA-COMPACT: Qualität mit Rechneranschluß

Typenradschreibmaschine mit Universal-Interface (c't-Februar-Heft), kann parallele oder serielle Daten verarbeiten (nur DIL-Schalter umschalten und Stecker austauschen!). Die **Halbzeilenschaltung** erlaubt die **Darstellung mathematischer Formeln (Matrizen, Potenzen).** Typenräder mit verschiedenen Schriften und Formelzeichen sind verfügbar.

Technische Daten:

- — parallele Schnittstelle (kann z. B. anstelle des MX-80 angeschlossen werden)
- — serielle Schnittstelle mit 1200 Baud (paßt z. B. an den Osborne- oder Zenith-Rechner) mit DTR-Handshake (DTR-Pegel wählbar)
- — deutscher Zeichensatz, Datenpuffer mit 44 Zeichen
- — Schreibgeschwindigkeit max. 14 Zeichen/s, 9 Zeichen/s inkl. Warenrückkauf

NEU: Interface-Version 3.2 mit wesentlichen Verbesserungen

Baugruppe mit paralleler Schnittstelle (jetzt auch Acknowledge-Signal verfügbar), RS-232-Schnittstelle mit DTR-Handshake und wählbarem CBM-Zeichensatz (z. B. C-64, VC-20). Wahlweise RS-232-Stecker bzw. Stecker für parallele Schnittstelle, SYCOM 3.2B ..... **DM 348,—**

wie vor, anstelle des CBM-Zeichensatzes XON-XOFF-Protokoll für die RS-232-Schnittstelle, SYCOM 4.0B ..... **DM 348,—**

CBM-Benutzer (2000-, 3000-, 8000er-Serie):
wie vor, für den CBM-IEC-BUS mit CBM-Zeichensatz SYCOM 1.0B ... **DM 398,—**

wie vor, jedoch als Bausatz SYCOM 3.2BS, SYCOM 4.0BS ..... **DM 298,—**

wie vor, jedoch nur Platine & CPU SYCOM 3.2P, SYCOM 4.0P ..... **DM 198,—**

Anschlußleitungen für C-64, VC-20 mit aktivem Interface ..... **DM 69,—**

Anschlußleitungen für Zenith Z 89/90, Osborne, Acorn ..... **DM 49,—**

Es liegen allen Baugruppen und Bausätzen Einbauhinweise, eine Bedienungsanleitung und der Zeichensatz bei.

OLYMPIA-COMPACT Typenradschreibmaschine, komplett mit Interface nach Wahl
(SYCOM 3.2BS, SYCOM 4.0BS) ..... **DM 1398,—**
mit Interface für CBM-IEC-BUS, SYCOM 1.0 ..... **DM 1448,—**

c't-Sprachsynthesizer:

Sprachausgabe aus jedem Rechner mit einer parallelen Schnittstelle. Freier Textaufbau durch Phonemsynthese, notwendige Datenrate ca. 10 Byte/s. Programmierbar über BASIC, PASCAL, C oder Assembler. Die Sprachausgabe nach dem c't-Projekt aus dem Februar-Heft.

Sonderausführungen für RS-232 und CBM-IEC-Bus sind lieferbar.

Fertigkarte (parallele Eingabe), Anschluß an VC-20, C-64, CBM 2001..., (über den USER-Port) und jede parallele Druckerschnittstelle möglich,
entsprechende Beschreibung wird mitgeliefert: .. **DM 298,—**

Wie vor, Bausatz: ..... **DM 248,—**

Wie vor, jedoch nur Sprachsynthesizer IC und Platine: ..... **DM 172,—**

Komplettsystem im Gehäuse mit Netztrafo, anschlußfertig: ..... **DM 448,—**

Wie vor, für CBM-IEC-BUS (z. B. CBM 2001,...) oder RS-232-Schnittstelle, anschlußfertig: ..... **DM 548,—**

Einkartenrechner mit Z80-CPU und Monitor

Für Meß- und Regelsysteme, Steuerungen.

- — ZE-2: 2 Z80 PIO's, Z80-Interrupt (IM 2), bis 8K-EPROM, 3K-RAM; 2,5 MHz Takt: ..... **DM 298,—**
- — ZE-64: wie ZE-2. zus. Z80A CTC, bis 16K-EPROM, 64K-RAM, 4 MHz Takt: ..... **DM 598,—**
- — AD 12/8: passender 8-Kanal-12-Bit-Umsetzer (50 µs/Kanal): ..... **DM 998,—**
- — AT-2: Anzeige und 24 Tasten passend zu ZE-2 & 64: ..... **DM 168,—**
- — ZE-2-SER: Serielle & parallele Schnittstelle für ZE-2 & ZE-64: ..... **DM 98,—**
- — SYS-2/64: Monitorprogramm für ZE-2/64, unterstützt ADU, Schnittstelle, Anzeige und Fehlersuche. Mit ZE-2/64 Hardwarebeschreibung, Monitorbeschreibung, **Listing** und Monitor im 2716-EPROM: ..... **DM 98,—**

Ab April lieferbar:

Datenerfassung mit dem VC-20/C-64
Grundeinheit mit Eingangsbuffer für maximal 8 A/D-Umsetzer, Datenerfassungsprogramm auf Kassette.
Die Grundeinheit wird an den USER-Port angeschlossen. ..... **DM 98,—**

Analog-Digitalumsetzer für den VC-20/C-64
Eingänge für Temperaturfühler (LM 390), ±1 V, ±1 mA sind vorhanden, Auflösung 2000 Schritte, eine Messung pro Sekunde, maximale Entfernung vom Rechner 50 m, 3adrige Zuleitung ..... **DM 98,—**

Referenzspannungsquelle

Digital einstellbare Gleichspannungsquelle 10 V in 10000 Schritten zu 1 mV, kurzschlußfest. Restwelligkeit unter 100 Mikrovolt.

Ideal zum Abgleich und zur Kontrolle von Analog-Digital-Umsetzern und als preiswerte Referenzquelle am Arbeitsplatz ..... **DM 398,—**

Auftragsentwicklungen: analoge und digitale Schaltungen.
Mikroprozessoren: Z80, 8085, MCS-48 und 6801/68701.
Speicherprogrammierte Steuerungen.

Alle Preise einschließlich Mehrwertsteuer, Versand per Nachnahme oder Vorkasse. Lieferzeiten 2...4 Wochen.

SYSTECH GmbH, Gliesmaroder Str. 26, 3300 Braunschweig, Tel.: 05 31/34 56 41 & 37 23 91

Interface für Apple II

Die Firma digitanalog stellt das universelle Interface 'OWIE' für Apple II-Rechner vor. Es arbeitet 'slotunabhängig' und erlaubt die parallele Dateneinund -ausgabe. OWIE verwendet CMOS-RAMs, die durch EPROMs ersetzt werden können, wodurch eine anwendungsspezifische Schnittstelle realisierbar ist.

Informationen: digitanalog Ewald Balfer, Hauptstr. 159, 8752 Krombach

Speichersystem für Multibus

Das multibuskompatible Speichersystem SWT80172 besteht aus einem Winchester-Laufwerk (Kapazität formatiert 11,2 MByte) und einem 8" Slim-Line Floppy-Laufwerk (Kapazität formatiert

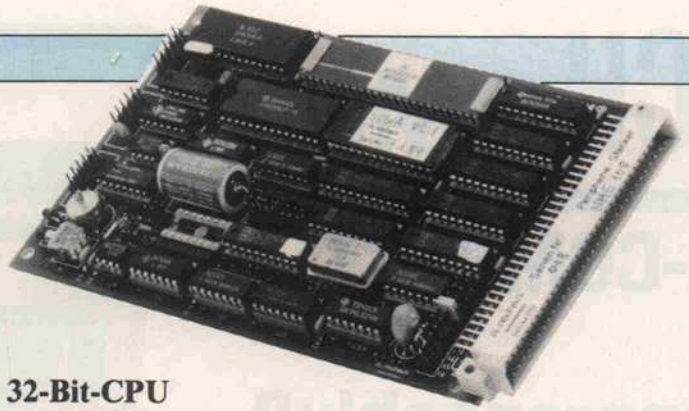
1,2 MByte). In dem flachen Gehäuse sind neben den Laufwerken noch ein Netzteil und ein Controller FWD8006 untergebracht. Die I/O-Adresse kann 8 Bit oder 16 Bit breit sein.

Informationen: SYNELEC Datensysteme GmbH, Lindwurmstr. 117, 8000 München 2

'Computern' im Urlaub

In den Monaten Juli, August und September findet auf der Insel Borkum die Computer-Sommer-Schule statt. In Gruppen mit maximal 12 Teilnehmern werden Themen wie 'Kleincomputerwissen für Einsteiger', 'Textverarbeitung' und 'BASIC' behandelt, wobei der Schwerpunkt auf praktischen Übungen liegt. Die Kurse dauern ein bis sechs Tage, an denen jeweils sechs Stunden unterrichtet wird.

Informationen: Ing.-Büro Auffarth, Rhalandstr. 54B, 2730 Zeven, Tel. 04281/1577



32-Bit-CPU am ECB-Bus

NME 108 ist die Bezeichnung für eine neue Zentraleinheit mit der 32-Bit-CPU 68008. Die Baugruppe ist für den Betrieb am ECB-Bus konzipiert und kann problemlos in vorhandene Systeme eingebaut werden. Die Karte bietet zwei V24 Schnittstellen und zwei 28polige Byte Wide Sockel. Diese ZE kann als 'Busmaster' arbeiten, ihr Adressierbereich beträgt 1 MByte. Preis: 1495 D-Mark (ohne Mehrwertsteuer).

Informationen: Dr. Neuhaus Mikroelektronik KG, Hochallee 39, 2000 Hamburg 13

Multi-User mit Commodore

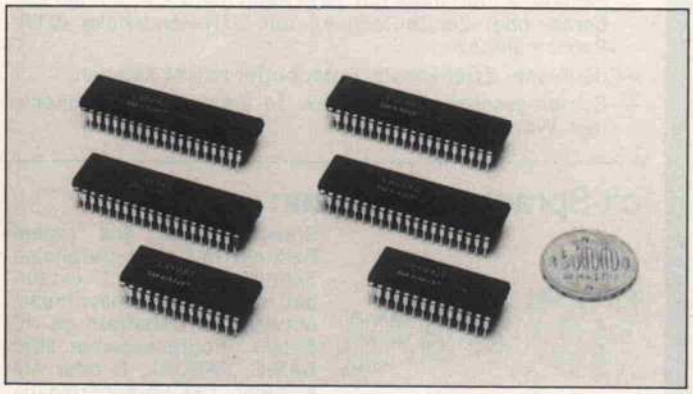
Das Multi-User-System 'Priority-Control' steuert den Zugriff von maximal 32 Commodore Rechnern der Serien 3000 bis 8000 und 700 auf ein gemeinsames Datenbanksystem. Als Speichermedium dient entweder die Commodore Floppy-Disk oder eine Hard-Disk. Das System ist kompatibel zu bestehender Commodore-Hardware und BASIC-Software. Für zwei CBM-Computer 8032/SK kostet das Grundsystem 1600 D-Mark (ohne Mehrwertsteuer).

Informationen: Johann F. Beurer, Victor-Achard-Str. 11, 6380 Homburg vdH.

Z80-Familie in CMOS

Die Z80-Bausteine werden nun von der Firma SHARP in CMOS-Technologie gefertigt. Die Schaltkreise haben bei einer Taktfrequenz von 2,5 MHz nur eine Stromaufnahme von 10 mA. Die ICs LH5080 (CPU), LH5081 (PIO) und LH5082 (CTC) sind zu der Z80-Serie kompatibel.

Informationen: REIN Elektronik GmbH, Lötscher Weg 66, 4054 Nettetal 1



Winchester für Mikros

Die 'Power-Box' von Ampex ist ein Winchester-Speicher für IBM PC, Apple und alle Computer, die mit Multibus, Q-Bus oder S 100-Bus arbeiten. Mit

der 5¼"-Platte stehen dem Anwender bis zu 22 MByte Speicherplatz zur Verfügung. Neben dem 'Host-Adapter', der in einen Einschub des Rechners gesteckt wird, benötigt man noch Steuerungssoftware, die mitgeliefert wird.

Informationen: Ampex Europa GmbH, Walter-Kolb-Straße 9, 6000 Frankfurt 70

Monitorlisting für c't 86

Das Monitorlisting für den 16-Bit-Rechner c't 86 liegt jetzt in gedruckter Form (39 Seiten) vor. Es kann gegen einen Kostenbeitrag von 5 DM beim Heise-Software-Service, Postfach 2746, 3000 Hannover 1, bezogen werden.

TI-Special II

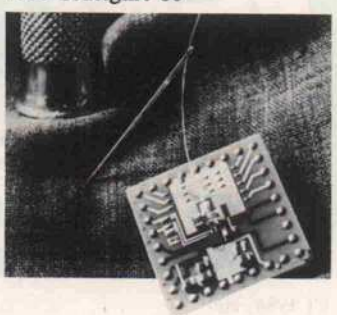
Für die Benutzer des TI 99/4A von Texas Instruments erscheint in Kürze das Handbuch '99 Special II'. Nach Angaben von TI bietet das neue Buch eine Steigerung gegenüber seinem Vorgänger 'Special I' und stellt 'hohe Anforderungen an Anwender und Maschine'. '99 Special II' ist im Buch- und Fachhandel erhältlich und kostet 54 D-Mark (incl. Mehrwertsteuer).

Informationen: Fachbuchvertrieb Wichmann + Partner, Geiselgasteigstraße 120, 8000 München 90

Kommunikation beschleunigt

Bei IBM wurde ein neuer Empfängerbaustein für Glasfaserkommunikationssysteme entwickelt. Der Chip erlaubt einen 16mal schnelleren Datenaustausch zwischen Rechner und Peripherie als die bisher verwendeten IBM-Verstärker. Daten können von dem neuen Chip mit 400 Megabit/s übertragen werden. Der Baustein fungiert gleichzeitig als Datenübertragungsverstärker und Empfänger.

Informationen: IBM Deutschland GmbH, Postfach 80 08 80, 7000 Stuttgart 80



Der professionelle Heimcomputer
ORIC-ATMOS

ORIC-ATMOS ist die technische Weiterentwicklung des ORIC-1, dem „Computer des Jahres 1983" in Frankreich.

- 64 K RAM
- CENTRONICS-Drucker-Schnittstelle
- 40 Zeichen x 28 Zeilen
- Grafik 240 x 200, 8 Farben
- 8 Oktaven, 3-Kanal-Synthesizer
- HI-FI-Ausgang

Weitere Informationen? Kein Problem, kostenlose INFO anfordern!

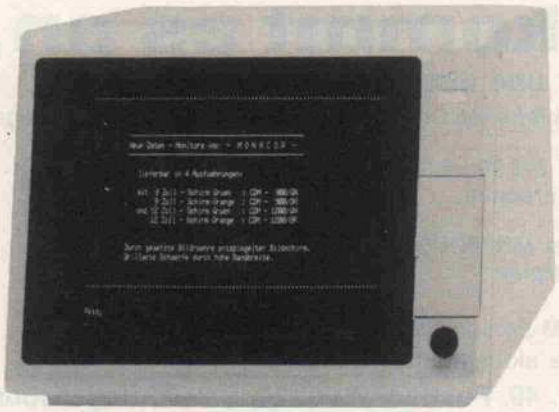
Oric-Atmos
Incl. Netzteil, Anschlußkabel für handelsüblichen Kassettenrecorder und Fernseher, Demo-Kassette und **deutschen** Handbuch ..... **748,—**

Oric MCP-40 Colour Printer
Ball-Point-Pen 4-Farb-System (schwarz, blau, grün, rot), voll grafikfähig, eingebaute Centronics-Schnittstelle, incl. Verbindungskabel und Handbuch ..... **698,—**

Oric Micro-Disc
3"-Disketten-Laufwerk mit 320 K Speicherkapazität (formatiert), incl. Controler und Handbuch ..... **1195,—**

Data-Recorder MVB-100
eingebauter Lautsprecher, Audio-Monitor-Schalter, 7-Pin-DIN-Buchse, Jackplug-Input-Buchse 3,5 mm, Jackplug-Output-Buchse 3,5 mm, Remote-Control-Buchse 2,5 mm, Lautstärkeregler, Bandzählwerk; Stromversorgung: 6-V-Batterien oder Netzteil; Übertragungsgeschwindigkeit: 300—2400 BAUD; Anschlußmöglichkeiten: ORIC-1, ORIC-ATMOS und andere Heimcomputer **99,50**

Daten-Display-Monitore



Auflösung: Horizontal/Vertikal 1000 Zeilen
Bandbreite: 22 MHz
Anzeige: Grün oder Orange
CDM-900/GN, 9" grün ..... **349,—**
CDM-900/OR, 9" orange ..... **369,—**
CDM-1200/GN, 12" grün ..... **359,—**
CDM-1200/OR, 12" orange ..... **399,—**

675	**Zubehör** 115 mm breit für MCP-40	
774	Joystick	35,—
775	Interface für Joystick	29,—
622	Universal-Netzteil für Data-Recorder	22,—
707	Druckerkabel Centronics	86,—
755	4-Farbstift-Set für MCP-40	15,—
776	Data-Kassetten C10	4,10
779	ORIC-Owner, Zeitschrift für ORIC-Besitzer (engl.)	5,50

Kombi-Angebot
Oric-Atmos + Oric Micro Disk ..... **1849,—**
Oric-Atmos + CDM-900/GN ..... **1049,—**
Oric-Atmos + MUB-100 ..... **799,—**

ABS(—4) is 4
ABS(4) is 4
A = ASC(N$)
Z = ATN(Y/4)
CALL X
CHAR X,S,FB
CHR$(value)
CIRCLE R,FB
CLEAR
CLS
CLOAD "XX"
CONT
COS
CURMOV
CURSET
CSAVE
DATA
DEEK
DEF FN
DEF USR
DIM
DOKE
DRAW
END
EXP
EXPLODE
FALSE
FILL
PRINT FNA(X)
FOR N = 1 TO NEXT N
FRE(0)
GET A$
GOSUB 1000
GOTO 4000
GRAB
PRINT HEX$(V)
HIMEM # 8700
HIRES
IF/THEN . . . ELSE
INK N
INPUT N$
INPUT "Age?";A
X = INT(Y + 0.5)
X$ = KEY$
L$ = LEFT$ (A$,N)
A = LEN(N$)
LET A = 4
LIST 100
LIST
LIST 50-80
LLIST 100
LLIST
A = LN(—2)
B = LOG(Y + 1)
LORES 0
A$ = MID$ (Z$,A,L)
MUSIC
NEW
ON N GOSUB 2000,3000
ON N GOTO 1000,2000
PAPER N
PATTERN X
A = PEEK(X)
PRINT 2+PI
PING
PLOT X,Y,"X"
PLOT X,Y,A$
POINT(X,Y)
POKE N,V
POP
A = POS
PRINT "HELLO"
PRINT N;A$
PULL
READ A$,N
REPEAT
REM IGNORE THIS
RETURN
RELEASE
RESTORE
R$ = RIGHT$ (A$,N)
A = RND(1)+6
RUN 200
SCRN(X,Y)
Z = SGN(X—Y)
SOUND
SHOOT
A = SIN(N)
PRINT "HO"
SPC(N) "HUM"
A = SQR(N)
STOP
N$ = STR$(N)
PRINT TAB(N) "HELLO"
A = TAN(N)
TEXT
TROFF
TRON
TRUE
USR(N)
A = VAL(N$)
WAIT(N)
ZAP

Software für Oric-ATMOS und Oric-1

Art.-Nr.	Spiele	
727	Hopper	29,50
716	Invaders	29,50
729	Harrier Attack	29,50
719	Oric-Munch	29,50
718	The Ultra	29,50
721	Light Cycle	29,50
734	Ultima-Zone	29,50
736	Rat Splat	29,50
735	Defence-Force	29,50
720	Acherons-Rage	29,50
722	Galaxians	29,50
723	Super-Meteors	29,50
724	Mushroom-Mania	29,50
728	Draculas-Revenge	29,50
783	Nowotnik Puzzle	29,50
751	Zorgons Revenge	35,—
737	Zodiac	35,—
715	Xenon-1	35,—
738	Oric-Chess	35,—
739	House of Death	35,—
714	Software Package (4 Kassetten) Oric-Flight Multigames I (5 Spiele: Bandit, Projectiles, Colourmatch, Quest, Reversi) Teach Yourself BASIC Home Finance	49,—
740	The Hobbit	55,—
741	Super Advanced Break out	19,50
731	Flight	19,50
732	Multigames I	25,—
752	Reverse	25,—
733	Multigames II (5 Spiele)	25,—
725	Dinky Kong	29,50
750	Candyfloss/Hangman	29,50
717	Centipede	29,50

Beruf und Betrieb

784	Languages: English, Spanish, Italian, German, French	55,—
742	Oric Calc (Spreadsheet)	55,—
743	Oric Base (Dateiverwaltung)	55,—
744	Author (Textverarbeitung)	55,—

Hilfsprogramme

745	Oric-Mon (Disassembler)	29,50
746	Oric-Mon (Ass./Disass.)	55,—
748	Oric-Forth (With Manual)	55,—

(alle Preise incl. MwSt.)

BESTELLCOUPON

Hiermit bestelle ich ☐ per Vorausscheck
☐ per Nachnahme (zuzügl. Nachnahmegeb.)

Stück	Art.-Nr.	Preis

Name _____
Straße _____
PLZ/Ort _____
Datum _____ Unterschrift _____

Werner Thoma · Auerstr. 29 · 7918 Illertissen
Telefon (0 73 03) 76 90

Software: Branchenpaket „Reisebüro" — Textverarbeitung „Fibu" — u. a.
Hardware: Die professionelle Konfiguration zu Ihrem Problem
Service: Auslieferung/Wartung max. 24 Std. bundesweit
Preis: Ob Händler oder Enduser, rufen Sie uns an

STA

S.T.A. Data Control Corp.
Türmergasse 25 · 6900 Heidelberg
Telefon 06221-780555

Für jeden den Richtigen
NEC
PC-8800

Bei uns auch die compl. NEC Druckerpalette erhältlich.
Passende Branchen-Software lieferbar.

Händleranfragen erwünscht

V. Linde elektronic
Neue Straße 18 · 7170 Schwäb. Hall · Tel. 0791/7318 · Telex 791102

Das Genie-Computer-Programm
APPLE II Kompatible 48 K Rechner

zu Superpreisen z. B.

Genie I oder II 64 K	1098,— DM
Colour Genie 16 K	598,— DM
48 K Computer	1148,— DM
Gemini 10X	1098,— DM

RK Computer-Electronic
Reinhard Koch
Bahnhofstraße 9, 3540 Korbach, Telefon 05631/63240

Auf die Software kommt es an,
auf das Know-how, den Support und auch den Preis!

CP/M-80, CP/M-86 und PC-DOS (MS DOS)-Software in ca. 50 verschiedenen Diskettenformaten in der Regel ab Lager, z. B.:

- Betriebssysteme: Concurrent CP/M-86, CP/M IBM PC/XT (DM 212,99 incl. MwSt.)
  Programmiersprachen: alle Microsoft und Digital Research-Sprachen
- Datenbanksysteme: KnowledgeMan, dBASE II, Friday!
- Textverarbeitung: WordStar, MS-WORD, in Verbindung mit der MS-MOUSE
- Tabellenkalkulationsprogramme: SuperCalc, Lotus 1-2-3, Multiplan
- und rund 300 andere Softwareprodukte

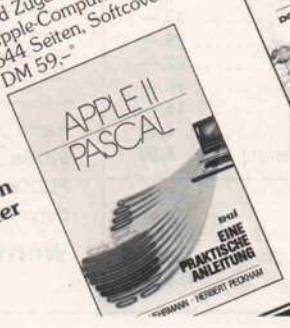
Erkundigen Sie sich auch nach unserem speziellen IBM Personal Computer-Hardwareangebot!
Fordern Sie Informationen und unsere aktuelle Preisliste an:

BSP Thomas Krug, Soft- und Hardware, Weißenburgstr. 49, Postfach 110324, D-8400 Regensburg
Tel. 0941/51945 und 51866, Tlx 652510

te-wi aktuell...

APPLE II – Anwenderhandbuch
(L. Poole)
Erst mit Hilfe dieses Leitfadens werden Sie Ihren Apple II erfolgreich einsetzen, denn Text und Bildmaterial gehen weit über das hinaus, was hersteller-seitig an Literatur angeboten wird. 416 Seiten, Softcover, DM 56.–*

APPLE II PASCAL – Eine praktische Anleitung
(A. Luehrmann, H. Peckham)
Unentbehrlich für alle, die die Programmiersprache PASCAL lernen wollen und Zugang zu einem Apple-Computer haben. 544 Seiten, Softcover, DM 59.–*



Alles für Ihren APPLE-Computer

APPLE MASCHINENSPRACHE
(D. Inman, K. Inman)
Dieses Buch bildet die einfachste Brücke zwischen dem vertrauten BASIC und der Maschinensprache des Mikroprozessors 6502 in Ihrem Apple-Computer.
224 Seiten, Softcover, DM 49.–*

* Die Preise sind die Ladenpreise

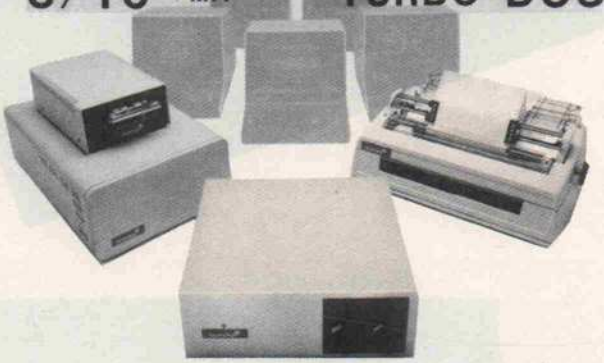
te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40
te-wi

tm 3812

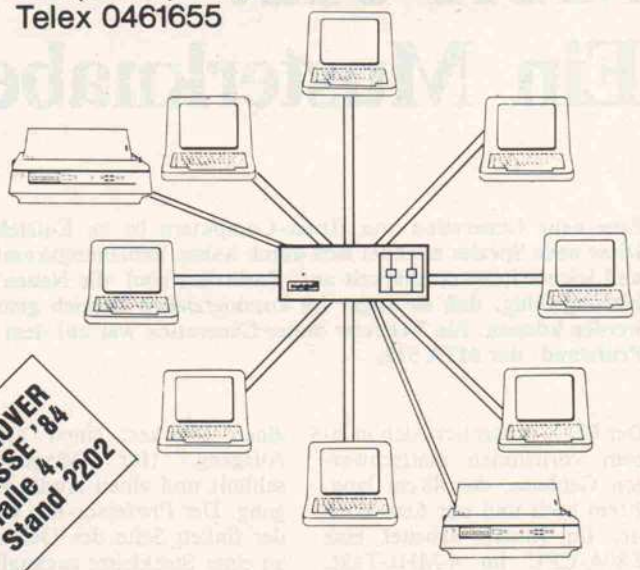
NorthStar™

Syscom

NORTH STAR HORIZON
8/16 MIT TURBO-DOS



Winkler GmbH
Trübnerstraße 40
6900 Heidelberg 1
Tel. (06221) 4 91 81
Telex 0461655



Einer der zuverlässigsten Mikro-Computer für Schule, Büro, Wissenschaft, jetzt noch bestechender in Leistung und Vielseitigkeit durch:

- Multiprocessing — bis zu acht Benutzer, jeder mit eigener Zentraleinheit
- Northstar-Betriebssystem Turbo-DOS
- 8-Bit- und 16-Bit-Zentraleinheiten
- Große Speicherkapazität — bis zu 512 K-RAM pro 16-Bit-Zentraleinheit

HANNOVER MESSE '84 Halle 4, Stand 2202

Das HORIZON 8/16 Multi-Processor-System unterstützt bis zu 8 Benutzer.

Achtung!! VC-20/VC 64!!

Wir haben alles für Ihren Computer!! Über **1000 Programme** aus allen Bereichen! **Schon ab 0,50/1,—/1,90.....! Internationale Software...Textverarbeitung... Dateiverwaltung** Utilitys..! **Komplette Programmpakete schon ab 3,—...5,—...8,—**...und...und...und!!

STOP | **Dieser Katalog mit über 100 Seiten wartet auch auf Sie!**

Der neue VC 20/64 Katalog

- Jetzt mit Profiinfo!
- PRO.PLAN das komplette Büro in High Res und Graphicssteuerung.
- Sprite und Graphic leicht programmiert (Listing)
- Einstieg in die Maschinensprache
- Superspiele
- Tabellen und Programmierformulare
- Lehr- und Lernprogramme
- Programmanleitungen und vieles mehr...

Mit vielen Routinen und Listings zum Eintippen.

Der Knüller!

Nicht nur Katalog, sondern auch ein **Informationswerk** für den **Anfänger und Fortgeschrittenen.** Hier finden Sie ... **Tabellen** ... **Tips und Tricks** ... **Detaillierte Programmbeschreibungen** ... **Leseproben** ... **Bauanleitungen** ... **Formulare** ... **Utilitys** ... **Programme zum Eintippen** ... **Die Frageecke** ... **Das Profiinfo** ... und ... und ... und ...

Sichern Sie sich heute noch Ihr persönliches Exemplar!

Ti 99/4A

ACHTUNG! Ab sofort! Ein umfangreicher Katalog mit **vielen Informationen, Tips, Tricks und Programmbeschreibungen** wartet auch auf Sie. TI 99/4a **Superspiele, Dateiverwaltung, Programmpakete Action** und **Adventure Games** sind nur ein kleiner Auszug aus unserem **umfangreichen Angebot.** Greifen Sie zu! Lassen Sie sich überraschen! **Eine Gratiskassette wartet auch auf Sie!** Und natürlich auch hier ... SUPERPREISE (Programme **ab 1,— ... 1,50 ... 2,—, Pakete ab 6,— ...) und, und, und!**

Achtung! Für 2,— DM (Porto oder Münze) senden wir Ihnen unseren neuesten VC 20/64 Katalog mit über 100 Seiten!! (TI 99/4a 0,80 DM). Computertyp nicht vergessen! Schreiben Sie uns heute noch!

S + S Soft
J. Schlüter
Schöttelkamp 23a
4620 Castrop-Rauxel 9

DAS SUPERDING
Klangwunder in Digitaltechnik

„Digital" ist zum Markenzeichen höchster Perfektion geworden. Neueste HiFi-Systeme, Tonträger etc. sind in dieser Technik ausgelegt, denn keine andere, derzeit bekannte Art der Informationsübermittlung ist störungsfreier, klarer und brillanter als die Digitaltechnik.

So ist es logisch, daß Wersi sich dieser Technik bedient und die Digital-Orgel ALPHA DX 300 vorstellt. Und das im bewährten Wersi-Selbstbau-System.

Heute noch Informationsmaterial anfordern!

WERSI
Wersi Orgel- und Piano-Bausätze
Industriestraße 3E 5401 Halsenbach
Telefon (06747) 7131 · Telex 42323

- tausende naturgetreue Klangfarben
- alle Funktionen und Klangfarben frei programmier- und speicherbar
- durch Software-Änderung viele Orgelfunktionen veränderbar

ALPHA Digital DX 300

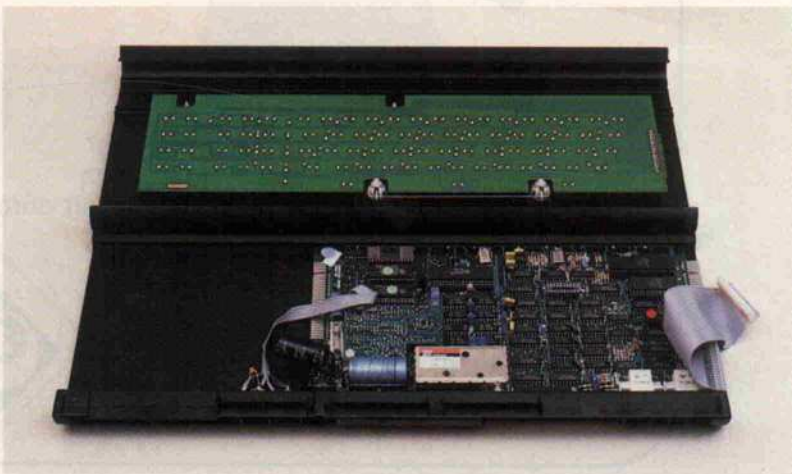
- alle Klangfarben auf jedes Manual, Pedal und Begleitautomatik schaltbar
- Rhythmusgerät mit digital abgespeicherten Instrumenten
- Begleitautomatik frei über Manuale programmierbar
- über Home-Computer spielbar
- mit M.I.D.I. und R 232 Schnittstelle
- extrem einfacher Selbstbau
- außergewöhnlich günstiger Preis

Gutschein
Gegen Einsenden dieses Coupons erhalten Sie ausführliche Informationen über die ALPHA-Digital und über den Orgel-Selbstbau. Bitte ausschneiden u. auf Postkarte kleben.

Andreas Burgwitz

MTX 512: Ein Musterknabe

Eine neue Generation von Home-Computern ist im Entstehen. Diese neue Spezies zeichnet sich durch hohen Bedienungskomfort und leichte Beherrschbarkeit aus. Zusätzlich sind 'die Neuen' so leistungsfähig, daß sie sogar im kommerziellen Bereich genutzt werden können. Ein Vertreter dieser Generation war auf dem c't-Prüfstand: der MTX 512.



Der MTX präsentiert sich in einem vornehmen mattschwarzen Gehäuse, das 48 cm lang, 20 cm breit und nur 6 cm hoch ist. Im Innern arbeitet eine Z80A-CPU im 4-MHz-Takt. BASIC und Betriebsprogramm belegen 24 KByte ROM, das (Farb-)Video-RAM umfaßt 16 KByte. Dem Anwender stehen in der Grundversion wahlweise 32 oder 64 KByte RAM zur Verfügung.

Wie viele Computer, die mit einer Z80-CPU bestückt sind, kann auch der MTX unter dem Betriebssystem CP/M laufen. Allerdings nicht in der Grundversion: Der Anwender muß nochmal tief in die Tasche greifen und das Floppy-Subsystem FDC (mit eigenem Gehäuse) zusätzlich anschaffen.

79 Tasten

Wir hatten ein Gerät mit deutscher Tastaturbelegung (siehe da!) im Test. Alle Tasten arbeiten mit Autorepeat. Neben den Standardtasten gibt es zwölf fest belegte und acht programmierbare Funktionstasten. Zwei weitere, unbeschriftete Tasten muß man drücken, um einen Reset auszulösen.

Ein Blick auf die Rückseite des MTX läßt eine fast verwirrende Anzahl von Steckerleisten und Buchsen erkennen. Neben den obligatorischen Anschlüssen für Fernseher (HF), Stromversorgung und Kassettenrecorder (zwei CINCH-Buchsen) bietet der MTX noch zwei Eingänge für Joysticks, einen Centronics-kompatiblen Anschluß für einen Drucker, einen Video-Ausgang (für Monitoranschluß) und einen Audio-Ausgang. Der Prozessor-Bus ist an der linken Seite des Gehäuses an einer Steckleiste zugänglich. Diese Anschlußmöglichkeiten dürften auch anspruchsvolle Anwender zufriedenstellen.

Ausbaufähig

Der MTX 512 trägt die Zahl hinter dem Kürzel als Hinweis darauf, daß er mit maximal 512 KByte RAM ausgerüstet werden kann. Erweiterungskarten kosten beispielsweise 675 Mark (128 KByte) oder 249 Mark (32 KByte).

Für weitere 269 Mark kann man sein MTX-System mit einem RS232-Interface ausrüsten. Das Interface ist auch zum Anschluß des FDX-Systems erforderlich. Man schafft sich damit außerdem die Möglichkeit, bis zu 255 MTX-Rechner zu einem Netz zusammenzuschalten.

Den Hobby-Anwender dürfte eher noch interessieren, daß neben dem Standard-ROM auch andere im Computer betrieben werden können. Einfache BASIC-Befehle ermöglichen die Umschaltung. Laut Hersteller sollen ROMs mit FORTH, Pascal, Textverarbeitungs-Software und Spielen erhältlich sein.

Das bereits erwähnte 'FDX-System 1' enthält neben einem Floppy-Controller und zwei 5¼"-Laufwerken auch eine 80-Zeichen-Farbkarte. Es kostet rund 3900 Mark. Zu teuer? — Abwarten: Zum Lieferumfang zählen das Betriebssystem CP/M 2.2 inclusive einiger 'Utilities', SuperCalc (Tabellenkalkulationsprogramm), ein an das MTX-BASIC angelehntes Disk-BASIC, ein relationales Datenbankprogramm (ähnlich dBase II) und ein Textverarbeitungsprogramm mit Namen 'New Word', das dem bekannten WordStar entsprechen, aber einfacher zu bedienen sein soll.

Bild und Ton

Nach jedem Einschalten teilt der MTX den Bildschirm automatisch in drei 'Zeilengruppen' ein: In einer 'Infozeile' erscheinen unter anderem die Fehlermeldungen. Vier 'Edit-Zeilen' bilden die gerade bearbeitete Programmzeile ab. Alles, was gelistet oder in den 'Edit-Zeilen' eingegeben wurde, stellt der Computer in einem 19 Zeilen umfassenden 'List-Schirm' dar. Wer glaubt, er könne die Ausgabe noch übersichtlicher gestalten, soll das tun: Passende BASIC-Anweisungen stehen zur Verfügung.

Initialisiert wird der Bildschirm im Text-Mode mit 24 Zeilen zu 40 Zeichen, im Grafik-Mode stehen 256x192 Bildpunkte in 16 Farben zur Verfügung.

Der schöpferischen Freiheit sind kaum Grenzen gesetzt. So stellt der MTX acht frei definierbare 'virtuelle Bildschirmseiten' bereit. Der Anwender bestimmt, was und in welchem Format eine Seite darstellen soll. Einmal definiert, kann jede Seite, zum Beispiel bei Programmstart, mit der anzuzeigenden Grafik 'beschrieben' werden. Durch einfaches Umschalten (durch das Programm) erreicht man so einen sehr schnellen Wechsel der Bilder.

Eine Reihe von BASIC-Befehlen unterstützt wirksam den Einsatz der grafischen Möglichkeiten. Es ist auch eine Art 'Schnittstelle' zwischen Grafik und BASIC vorhanden. So kann man zum Beispiel Bildschirmpunkte abfragen und deren 'Werte' unter Variablen ablegen.

Sollen sich die 'bunten Bilder' auch bewegen, empfiehlt sich die Verwendung von 'Sprites'. Der MTX bietet eine Sprite-Grafik, die ebenfalls durch BASIC Befehle recht einfach zu programmieren ist. Ein

Sprite besteht aus 8x8 oder 16x16 Pixeln (Bildpunkten). Man kann bis zu 32 einzelne Sprites unabhängig voneinander steuern und in einer Zeile maximal vier Sprites darstellen. Farbe und Bewegungsrichtung sind frei wählbar.

Für die akustische Untermalung sorgen drei Tongenerator-Kanäle und ein Rauschgenerator. Tonhöhe, Hüllkurve und Lautstärke des produzierten 'Sounds' werden durch die Parameter eines BASIC-Befehls festgelegt. Auch kann man bestimmen, ob nur eine Note bis zum Stop-Befehl oder eine Folge von Tönen gespielt werden soll. Um eine Notenfolge erklingen zu lassen, muß man einen 'Sound-Buffer' anlegen, in dem die Argumente für den Soundgenerator stehen.

Wer (oder was) ist NODDY?

```
Prog
        ^S*DISPLAY TEXT.
          *ENTER
          *IF A,L
          *BRANCH S
        ^L*DISPLAY TEXT1.
          *RETURN
```

Dieses Programm ist in NODDY geschrieben. Sie kennen NODDY nicht? Hinter dem seltsamen Namen verbirgt sich eine Mini-Programmiersprache des MTX, deren 'Wortschatz' sich auf zwölf Befehle beschränkt. NODDY soll leicht zu erlernen sein und ein einfaches 'Text-Handling' ermöglichen. Um mit NODDY die ersten Erfahrungen sammeln zu können, braucht man lediglich in BASIC das Wort 'NODDY' einzugeben.

NODDY arbeitet seitenorientiert; jede Bildschirmseite muß mit einem Namen versehen werden und kann entweder als reine Text-Seite oder als Programm-Seite dienen. Eine Text-Seite kann in Programmen aufgerufen werden, um beispielsweise längere Texte bequem im eingegebenen Format abzubilden. Eine Programm-Seite birgt, wie der Name sagt, ein Programm. Die zwölf NODDY-Befehle ermöglichen es tatsächlich, die grundlegenden Abläufe zu programmieren. Sprünge und Verzweigungen lassen sich ebenso realisieren wie Ein- und Ausgaben. NODDY-Programme lassen sich aus BASIC aufrufen. Für die 'ersten Schritte' als Homecomputer-Programmierer vielleicht ein ganz nützliches Hilfsmittel.

Das Beispielprogramm befindet sich auf einer Seite namens 'PROG'. Wird es gestartet, zeigt es den Text der Seite 'TEXT' an (DISPLAY TEXT.) und wartet dann auf eine Eingabe (ENTER). Gibt man den Buchstaben 'A' ein, erfolgt ein Sprung zu dem Label '^L' (IF A,L). Andernfalls wird 'TEXT' erneut gezeigt (BRANCH S). Wurde zu dem Label '^L' verzweigt, gibt das Programm die Seite 'TEXT1' auf den Schirm aus (DISPLAY TEXT1.) und springt dann zurück zu BASIC (RETURN).

Das BASIC des MTX 512 ist schnell und umfangreich (siehe Tabellen 1 und 2). Allerdings ist es auch recht 'pingelig', was die Syntax der BASIC-Befehle betrifft. So verweigert der MTX die Variablenzuweisung ohne die Verwendung des Wörtchens 'LET'. Ebenso genau nimmt er es bei einer 'IF...THEN' Konstruktion.

Rechner	Programm							
	1	2	3	4	5	6	7	8
TRS-80 Modell 100	3,7	9,8	26,6	29,7	31,4	46,8	62,8	30,9
Apple II Plus	1,4	8,4	15,8	17,6	19,0	28,4	45,0	10,4
VC-20	1,2	8,1	15,3	16,8	18,1	27,1	43,0	9,6
Dragon	1,2	8,6	17,0	18,0	19,5	28,9	42,3	10,9
TRS-80 Modell 1 LII	2,8	11,2	27,0	27,8	31,0	50,6	78,0	11,8
ORIC-1	2,3	17,8	29,7	32,0	39,2	53,2	79,2	12,7
C-64	1,2	9,4	18,2	20,5	21,4	32,1	51,1	11,3
alphaTronic PC	2,2	5,3	15,4	16,7	18,1	31,0	42,6	17,8
BBC-ACORN	0,7	2,9	7,9	8,4	8,8	13,5	20,9	4,8
EPSON QX-10	2,0	6,2	15,6	14,6	16,4	31,9	52,8	6,8
MTX 512	1,9	5,3	11,6	11,5	13,3	23,1	40,9	4,8*

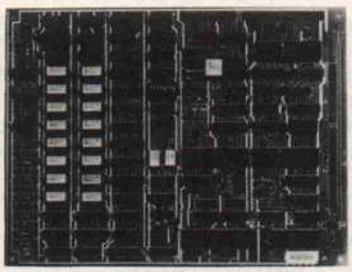
* mit natürlichem Logarithmus (die anderen Werte beziehen sich auf die Berechnung des Brigg'schen Logarithmus

Tabelle 1. Ergebnisse des Benchmark-Tests (Zeiten in Sekunden)

BULLET — das CP/M-Geschoß

Einer der schnellsten und leistungsfähigsten CP/M-Rechner für CP/M 3,0 (Plus) und 2,2

System-Lieferung möglich:
mit 2 mal 5,25-Zoll-Laufwerken je 1 MB oder 2 mal 5,25-Zoll-Laufwerken je 1,6 MB oder 5,25-Zoll-Laufwerk und eingeb. Harddisk 10 MB oder 2 mal 8-Zoll-Laufwerke DS/DD je 1,6 MB oder 8-Zoll-Laufwerk plus Harddisk ab 10 MB



20 x 27 cm

- CP/M-3,0 (Plus)-Software ist im Lieferumfang enthalten
- CP/M 2,2 mit 47 KB-RAM-Pseudo-Floppy möglich
- CP/M 3,0 mit 60 KB freiem Benutzerspeicher
- Z80 A — 4 MHz
- Durch DMA und Disk-Track-Buffering superschnell bei Disk-I/O: 64 KB werden in 14 sec mit Verify auf Disk geschrieben und in 10 sec gelesen, MBASIC in 3,3 sec gelesen (siehe Vergleichstest in BYTE Jan./Nov. 82)
- Anschluß für Parallel- und Seriell-Printer, Terminal Anschlußmöglichkeit für ECB-Bus-Karten, weitere V24, RAM-Floppy etc.

Universell einsetzbar für:
- OEM's
- Praxis und Hobby
- Entwicklung
- Universitäten und Institute

Fachhändler gesucht

- Verwaltung und Anschluß für vier Floppys 5,25-Zoll (40, 80 oder 77 Spur) und Anschluß für vier Floppys 8-Zoll (SS/SD oder DS/DD) gleichzeitig, Anschluß über Flachbandkabel Harddisk-Anschluß serienmäßig
- Kopieren der Software von 5,25-Zoll auf 8-Zoll und Kopieren der Software von 8-Zoll- auf 5,25-Zoll-Diskette
- alle gängigen Programmsprachen lauffähig
- Lieferung als Platine oder als Komplettsystem — Sie bestimmen, wir liefern —
- Kundenspezifische Applikationen möglich
- Anschluß-Platine für 8086 für MS-DOS und RAM-Floppy bis 1 MB vorgesehen.
- Convertierungsbullet zum Lesen und Kopieren von Daten fast aller Disketten anderer Computer.

Ausführliche Unterlagen und Anwendungsbeispiele bei:

EPS
M. Mandt
Ilmspaner Straße 29, D-6971 Großrinderfeld
Telefon (0 93 49) 2 71-12 71, Telex 6 89 549 EPS D

Eine Zeile, die hinter dem 'THEN' nur die Angabe der Zeilennummer enthält, wird einfach als falsch abgewiesen, der MTX fordert 'THEN GO-TO'. Die Syntax-Prüfung erfolgt, wie beim Sinclair-BASIC, gleich nach der Eingabe. Wird eine Zeile als falsch erkannt, bietet der Rechner sie sofort zur Korrektur an.

Wer die Werte von Variablen auf Kassette sichern möchte, muß das gesamte Programm speichern. Wer Programmzeilen und zeilenbezogene Befehle mit neuen Nummern versehen will, muß sich selbst an die Arbeit machen: 'Renumber' kennt der MTX nicht.

Von diesen kleinen Schwächen abgesehen, unterstützt das BASIC die Möglichkeiten des Computers ausgezeichnet. Erwähnenswert sind vor allem die Grafik- und Sprite-Grafik-Befehle. Außerdem ist der MTX mit einer Echtzeit-Uhr ausgerüstet, die ebenfalls durch BASIC-Befehle gestellt und abgefragt wird.

Wer sonst bietet das: Einen Homecomputer mit Assembler im ROM. Nicht nur eingefleischte Assembler-Programmierer werden dieses nützliche Werkzeug begrüßen, erlaubt es doch, bequem und ohne die Verwendung von 'Platzhalter'-Tricks, BASIC-Programme mit Routinen in Maschinensprache zu ergänzen. Der Assembler erwartet die Eingabe der üblichen Z 80-Mnemonics und ermöglicht die Verwendung von Kommentaren und symbolischen Sprungadressen (Labels).

Soll ein so erstelltes Programm getestet werden, kann man mit dem BASIC-Befehl 'PANEL' ein 'Debugger'-Programm aktivieren. Damit ist es möglich, einen Speicherinhalt in hexadezimaler oder mnemonischer Darstellung aufzulisten und gegebenenfalls zu verändern. Ein 'Single-Step'-Befehl erlaubt es, ein Programm Befehl für Befehl abarbeiten zu lassen und nach jeder ausgeführten Instruktion den Inhalt der Prozessorregister auf dem Schirm anzuzeigen.

Das gut gestaltete Handbuch lag zum Zeitpunkt des Tests leider noch nicht in deutscher Sprache vor. Es enthält ausführliche und anschauliche Beschreibungen der implementierten Befehle. Im Anhang finden sich die Schaltpläne des Computers und ein Auszug aus dem ROM-Listing. Wenn dieses 'Manual' ohne Abstriche am Inhalt in ein 'Handbuch' übersetzt wird, gibt es keinen Anlaß, daran etwas auszusetzen.

In Assembler

Fazit

Mit Preisen von 1198 Mark (32 KByte RAM) beziehungsweise 1390 Mark (64 KByte) für die Grundversion ist der MTX 512 nicht gerade billig. Aber dafür bietet er auch neben einer soliden Grundausstattung nahezu alles, was man sich bei einem modernen Homecomputer wünscht. Sowohl als System für den Einsteiger, der mit NODDY die ersten Programmierversuche unternehmen möchte, als auch für den fortgeschrittenen Hobbyisten erscheint der MTX empfehlenswert. Die vielfältigen Erweiterungsmöglichkeiten bis hin zum Bürocomputer sind gewiß ein weiteres Plus.

Befehlsvorrat MTX BASIC

ADJSPR	DIM	LPRINT	REM
ANGLE	DRAW	MSVPR	RESTORE
ARC	DSI	NEW	RETURN
ASSEM	EDIT	NEXT	ROM
ATTR	EDITOR	NODDY	RUN
AUTO	ELSE	NODE	SAVE
BAUD	FOR	ON	SBUF
CIRCLE	GENPAT	OUT	SOUND
CLEAR	GOSUB	PANEL	SPRITE
CLOCK	GOTO	PAPER	STEP
CLS	IF	PAUSE	STOP
CODE	INK·	PHI	THEN
COLOUR	INPUT	PLOD	TO
CONT	LET	PLOT	VERIFY
CRVS	LINE	POKE	VIEW
CSR	LIST	PRINT	VS
CTLSPR	LLIST	RAND	
DATA	LOAD	READ	

MTX Funktionen

ABS	NOT
AND	OR
ASC	PEEK
ATN	PI
COS	RND
EXP	SGN
INP	SIN
INT	SQR
LEN	TAN
LN	USR
MOD	VAL

NODDY-Befehlsvorrat

ADVANCE	IF
BRANCH	LIST
DIR	PAUSE
DISPLAY	RETURN
ENTER	STACK
GOTO	OFF

MTX String-Funktionen

CHR$	RIGHT$
GR$	SPK$
INKEY$	STR$
LEFT$	TIME$
MID$	

MTX Operanden

+	=
—	>
*	<
/	>=
^	

Tabelle 2.

Ergebnisse auf einen Blick

- ⊕ farbige Grafik mit Sprites
- ⊕ kurze Ausführungszeiten
- ⊕ Assembler/Disassembler und Debugger im ROM
- ⊕ Mini-Programmiersprache NODDY im ROM
- ⊕ sofortiger Syntax-Check
- ⊕ deutscher Zeichensatz
- ⊕ mit Erweiterung CP/M-fähig
- ⊖ Variable können nur zusammen mit dem Programm auf Kassette gespeichert werden
- ⊖ teilweise 'pingeliges' BASIC
- ⊖ kein Renumber

Centronics-Interface für den Spectrum 16/48 K

Andreas Stiller/Andreas Burgwitz

Nachdem c't in der letzten Ausgabe die Bauanleitung eines preiswerten Parallelinterface für den ZX 81 mit dem Portbaustein 8255 veröffentlicht hat, folgt nun die Version für den größeren Bruder. Dabei wird der 8255 nicht mehr 'memory mapped' adressiert, sondern über I/0. Daraus ergibt sich der Vorteil, daß dem Spectrum nicht irgendwo vier Speicherplätze verlorengehen, und daß das Interface auch über die BASIC-Befehle IN und OUT ansprechbar ist.

Wer allerdings glaubt, mit diesen komfortablen Befehlen alle 256 möglichen Portadressen nutzen zu können, der irrt. Der Spectrum decodiert nur die unteren fünf Bits für seine internen Ports (Kassetteninterface, Keyboard, Drucker, Microdrive). Diese fünf Bits müssen auf eins gesetzt sein, wenn man externe Ports ansprechen will. Die verbleibenden drei Bits erlauben dann noch, acht verschiedene Portadressen zu unterscheiden. Vier benötigt die PIO, die ja in der Bauanleitung für das ZX 81-Interface (siehe c't 4/84) schon ausführlich beschrieben worden ist. Das CHIP SELECT-Signal ($\overline{CS}$) für die PIO wird mit dem Binär-Decoder 74LS138 aus den Adreßleitungen A1 bis A4 (alle eins), A5 (auf null) und $\overline{IORQ}$ (aktiv low) gebildet. Die PIO selber decodiert A6 und A7 für ihre Ports und das Statusregister. Daraus ergeben sich die in der Tabelle wiedergegebenen Portadressen.

IC 2 verzögert $\overline{RD}$ und $\overline{WR}$ um eine Gatterlaufzeit gegenüber $\overline{CS}$, um ein korrektes Timing für die PIO zu ermöglichen.

Statt in BASIC ...

Die Initialisierung der PIO erfolgt beim Spectrum genauso wie beim ZX 81. Port B bleibt dem Benutzer zur freien Verfügung. Port A ist das Druckausgaberegister, und Port C wird gesplittet: die unteren vier Bits zur Eingabe (Bit 0 für $\overline{BUSY}$), und die oberen vier zur Ausgabe (Bit 7 für $\overline{DATA\ STROBE}$). Demnach erhält das Statusregister der PIO den Wert 129 (81h). Außerdem wird $\overline{DATA\ STROBE}$ auf eins gesetzt, um dem Drucker mitzuteilen, daß noch kein Zeichen vorliegt.

... lieber in Maschinensprache

Natürlich könnte ein BASIC-Programm die Druckeransteuerung übernehmen. Eine wesentlich elegantere Lösung läßt sich jedoch in Maschinensprache erreichen. Es kann nämlich in der Initialisierungsroutine (6000h ≙ USR24576) der Ausgabekanal des ZX-Druckers auf eine eigene Ausgaberoutine umgeschaltet werden, so daß die Druckbefehle LPRINT und LLIST verwendbar sind. Die Routine CODE prüft, ob CR (Carrige Return) oder ein Token vorliegt. Bei CR läßt sie ein LF (Line Feed) folgen; bei einem Token springt sie in die zuständige Übersetzungsroutine im ROM.

Der Spectrum speichert im Unterschied zum ZX 81 glücklicherweise alle Zeichen — bis auf die Grafiksymbole — in ASCII. Die Routine CODE übergibt daher Zeichen für Zeichen eines grafikfreien Strings ohne Umwandlung an das Druckausgabe-Unterprogramm PRINT. Dieses befragt in einer Schleife solange Bit 0 von Port C, bis der Drucker seine 'Bereit'-Meldung abgibt. Dann wird das aktuelle Zeichen an Port A ausgegeben und dies dem Drucker mit dem $\overline{DATA\ STROBE}$-Signal (Bit 7 von Port C kurzzeitig auf null) angezeigt.

Falls der Ausgabestring Farbkontroll- und Tabulatorbefehle enthält, interpretiert sie der Drucker als ASCII-Steuerzeichen. Um dem vorzubeugen, sollten man in CODE noch zusätzlich die Farbkontrollbefehle durch ein Fragezeichen ersetzen und Tabulatorbefehle gegebenenfalls durch spezielle Routinen behandeln.

```
6000            0010        ORG   6000H
                0020 ;            "Startadr 6000h"
                0030 ;            "Portstatus 129 (81h)"
6000 3E81       0040 INIT   LD    A,81H
6002 D3DF       0050        OUT   (0DFH),A
                0060 ;            "STROBE =1"
6004 3E80       0070        LD    A,80H
6006 D39F       0080        OUT   (9FH),A
                0090 ;            "definiert Channel"
                0100 ;            "mit Druckrout.6016h"
6008 2A4F5C     0110        LD    HL,(5C4FH)
600B 010F00     0120        LD    BC,000FH
600E 09         0130        ADD   HL,BC
600F 3616       0140        LD    (HL),16H
6011 23         0150        INC   HL
6012 3660       0160        LD    (HL),60H
6014 C9         0170        RET
6015 00         0180        NOP
6016 FE0D       0190 CODE   CP    0DH
6018 2809       0200        JR    Z,ENTER
601A FE80       0210        CP    80H
601C 380A       0220        JR    C,PRINT
                0230 ;            "print Token"
601E D6A5       0240        SUB   0A5H
6020 C3100C     0250        JP    0C10H
6023 CD2860     0260 ENTER  CALL  PRINT
                0270 ;            "ENTER ergibt CR und NL"
6026 3E0A       0280        LD    A,0AH
6028 F5         0290 PRINT  PUSH  AF
6029 F3         0300        DI
602A 3E7F       0310 BUSY   LD    A,7FH
                0320 ;            "wartet auf BUSY =0"
                0330 ;            "mit BREAK-Abfrage"
602C DBFE       0340        IN    A,(0FEH)
602E 1F         0350        RRA
602F 3011       0360        JR    NC,BREAK
6031 DB9F       0370        IN    A,(9FH)
6033 1F         0380        RRA
6034 38F4       0390        JR    C,BUSY
6036 F1         0400        POP   AF
                0410 ;            "Ausgabe A an Port A"
6037 D31F       0420        OUT   (1FH),A
                0430 ;            "DATA STROBE wird kurzzeitig"
                0440 ;            "auf 0 gesetzt"
6039 AF         0450        XOR   A
603A D39F       0460        OUT   (09FH),A
603C 3E80       0470        LD    A,080H
603E D39F       0480        OUT   (09FH),A
6040 FB         0490        EI
6041 C9         0500        RET
6042 CF         0510 BREAK  RST   8H
6043 0C         0520        DEFB  0CH
                0525        END
```

Über das Interface ausgegebenes Assemblerlisting

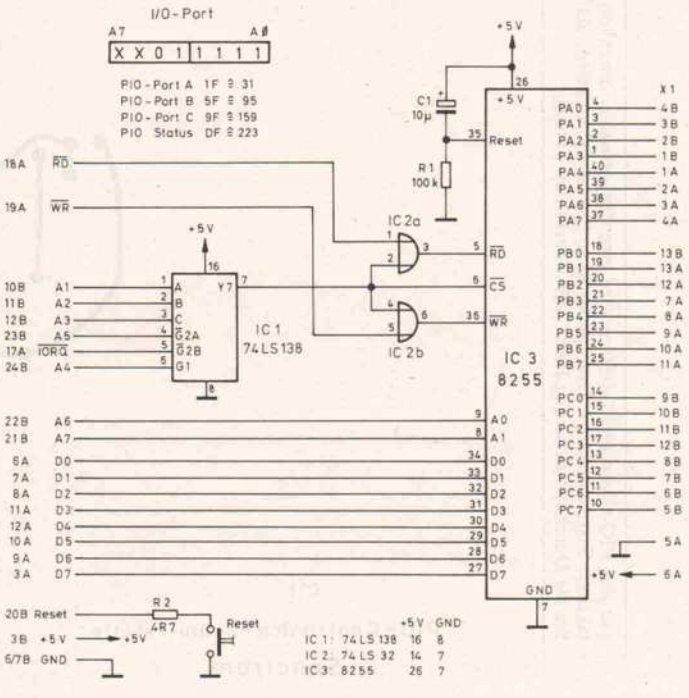
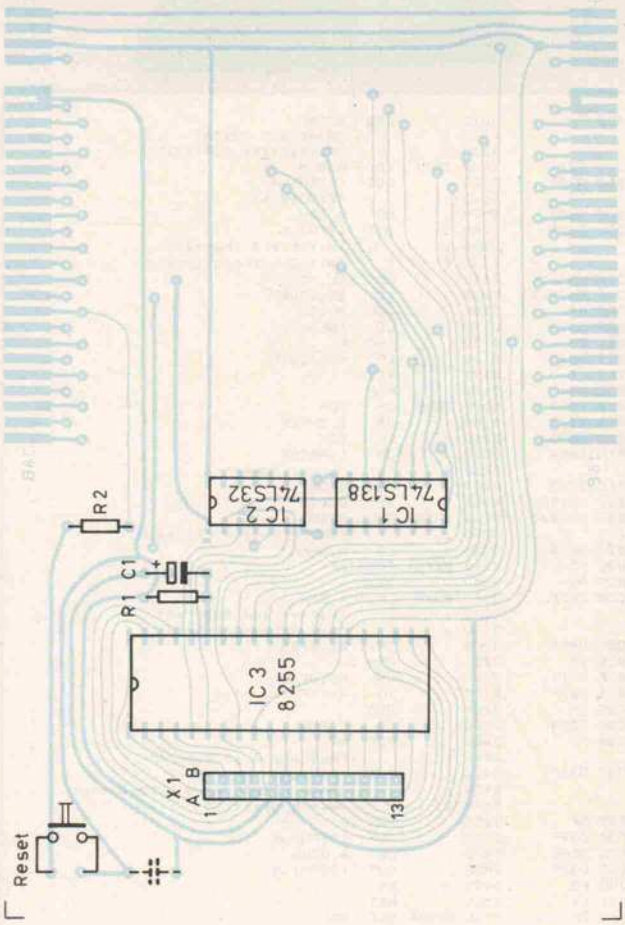
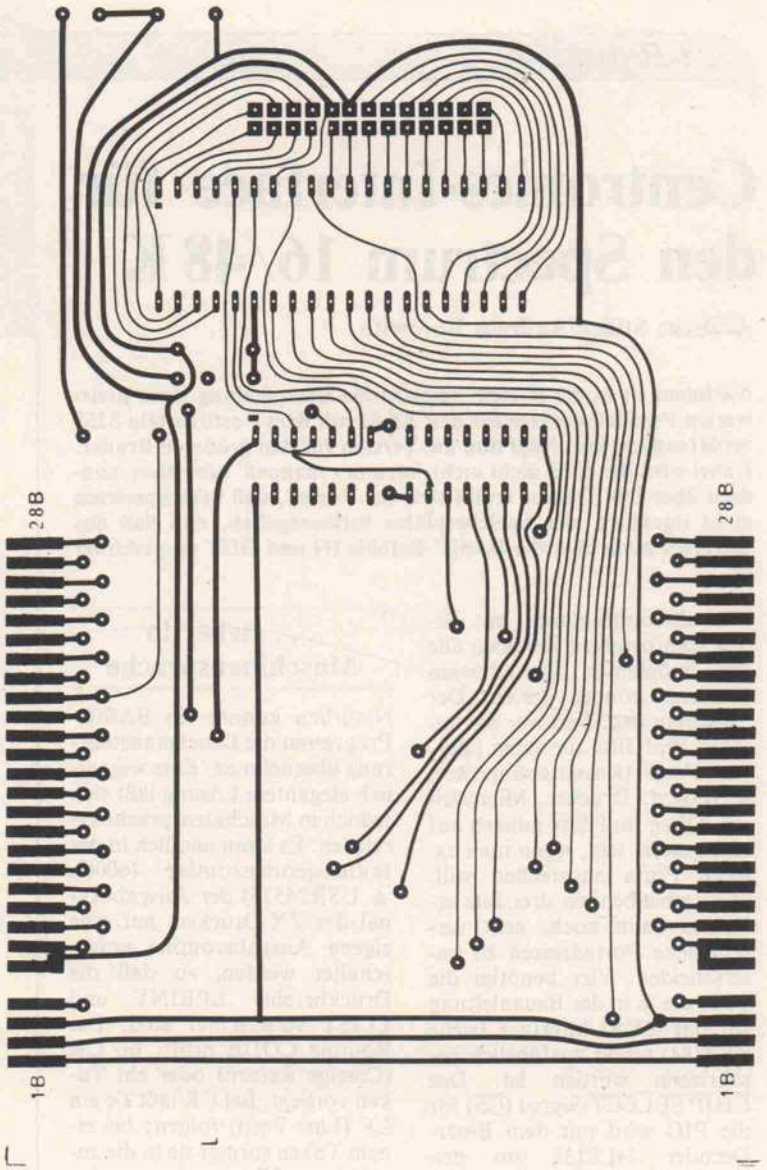
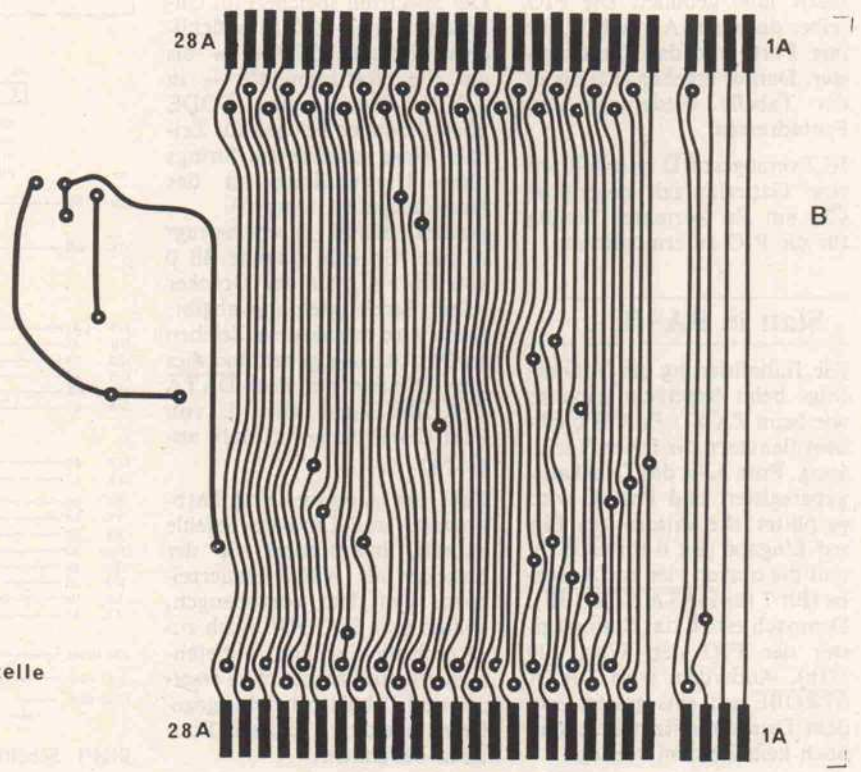


Bild 1. Schaltbild und Tabelle der Portadressen







Für das PIO-Centronics-Interface wird eine doppelseitige, durchkontaktierte Platine benötigt. Das Layout ist so 'geräumig' ausgelegt, daß Sie die Durchkontaktierung leicht selbst ausführen können.

c't
PIO-Centronics-Schnittstelle
Spectrum

Wenn der Mikro streikt

Triggerprogramme für die Fehlersuche mit dem Oszilloskop

Herbert Nabereit

Bei Schaltungen, die mit 'diskreten' Logik-ICs aufgebaut sind, kommt man bei der Fehlersuche ganz gut mit Oszilloskop und eigener Logik über die Runden. Versucht man aber, das emsige Schaffen eines 'entgleisten' Mikroprozessors mit einem Oszilloskop zu verfolgen, stößt man auf erhebliche Probleme: Wie im Fernsehen zeigt der Schirm an allen spannenden Stellen hauptsächlich Geflimmer. Bei neuentwickelten Systemen weiß man darüber hinaus noch nicht einmal, ob man eigentlich gegen die Hardware oder die Software kämpft. Unser Beitrag zeigt Ihnen, wie Sie dennoch ohne Logik-Analysator und In-Circuit-Emulator zu einem funktionierenden Prozessorsystem gelangen.

Das Hauptproblem bei der Untersuchung von Mikroprozessorsystemen mit dem Oszilloskop ist die Komplexität der Prozessorsignale. Dadurch ist es generell schwierig, stehende und lichtstarke Schirmbilder zu erhalten. Findet man tatsächlich ein zyklisches Signal, auf das man so triggern kann, daß aussagefähige Bilder auf den Schirm gelangen, liegt das interessierende Geschehen sicherlich außerhalb des auf dem Oszilloskop darstellbaren Bereichs. Es gilt daher, den Prozessor zur Mitarbeit heranzuziehen, was mit Hilfe einiger programmierter EPROMs, eines Oszilloskops (am besten zwei oder mehr Kanäle) und etwas Gedankenarbeit möglich ist. Die Vorgehensweise wird beispielhaft für Z80-Systeme erläutert, das Verfahren kann aber sinngemäß auf alle Prozessortypen angewendet werden.

Das Oszilloskop wird nur mit manueller Triggereinstellung betrieben, weil die automatische Betriebsart nicht sicher verwendbar ist. In vielen Fällen ist es erforderlich, den Takt als Referenz darzustellen, um die Impulse zuordnen zu können. Ein mehrkanaliges Oszilloskop bringt hier entscheidende Vorteile, ist jedoch in der Regel sehr viel teurer. Für die hier vorgestellten Anwendungen kann ein digitaler Multiplexer wie der c't-Scope Extender in vielen Fällen eine echte Hilfe sein.

Durch Verwendung einfacher und damit übersichtlicher Programme werden zyklische Impulsdiagramme erzeugt, so daß ohne spezielle Ausgaben (zum Beispiel über LEDs) die Funktion eines Mikroprozessorsystems auch dynamisch (um beispielsweise Zugriffszeiten zu ermitteln) mit einem Oszilloskop geprüft werden kann. Vorausgesetzt wird die Kenntnis der Prozessorhardware. Der vorliegende Beitrag kann allerdings auch als Ausgangspunkt für das Kennenlernen der Hardware typischer Prozessorsysteme dienen, da alle wichtigen Vorgänge recht anschaulich auf dem Oszilloskop darstellbar sind.

Was geht?

Bei neu entwickelten Systemen ist in der Regel zu erwarten, daß nach dem ersten Einschalten nichts passiert. Terminals, Tastaturen und auch Drucker arbeiten nicht. Der Prozessor beschäftigt sich mit unergründlichen Dingen oder macht gar nichts. Gleiches gilt für Systeme, die plötzlich ausfallen und den Anwender damit auch der Möglichkeit berauben, mittels Systemsoftware auf Fehlersuche zu gehen.

Ein grundsätzlicher Unterschied besteht zwischen Neuentwicklungen und fertigen Systemen. Bei Neuentwicklungen ist es fraglich, ob die Hardware überhaupt arbeitsfähig ist. Auch kann die Software noch fehlerbehaftet sein, so daß die Fehlersuche in mehrere Richtungen gehen muß. Bei defekten Systemen ist auf jeden Fall bekannt, daß das Grundkonzept arbeitsfähig ist. Aufwendige Untersuchungen der Logik und der Programme können daher entfallen.

Weiterhin sei darauf hingewiesen, daß bei fertigen Systemen der Austausch von Platinen oder Bauelementen bei der Eingrenzung eines Fehlers sehr hilfreich ist. Man sollte die wichtigsten Bestandteile schnell austauschen können, wenn der Fehler gar zu rätselhaft ist. Zeigt sich dann der gleiche Effekt wie vorher, so muß die Fehlersuche natürlich fortgesetzt werden.

Die vorgeschlagene Vorgehensweise trennt Hard- und Software-Untersuchungen. Hier wird versucht, die Funktion der Hardware sicherzustellen. Dies erfolgt mit wenigen, einfachen Standard-Prozeduren, die immer wieder verwendet werden können.

Für die Fehlersuche der Software gibt es eine Vielzahl von Verfahren, die sehr vom jeweiligen Einsatzzweck abhängen und daher hier nicht mitbehandelt werden.

Stehende Bilder

Zur Analyse der Prozessorfunktion ist es erforderlich, geeignete Triggerpunkte für das Oszilloskop zu schaffen, da sonst eine Beobachtung der Systemfunktion nicht möglich ist. Dazu eignen sich zum Beispiel I/O- oder Adreßaufrufe, die über den benutzten Adreßbereich hinausführen. Die entsprechenden Adressen dürfen daher nur von den Testroutinen aktiviert werden. Bei den 8080-Abkömmlingen stellt dies kein Problem dar, da die höheren Adressen weder gemultiplext werden noch mit Refresh-Signalen belastet sind.

Ebenso muß sichergestellt werden, daß das zu untersuchende Ereignis häufig genug auftritt, damit auf dem Bildschirm noch eine ausreichende Helligkeit erreicht wird. Die Routinen müssen daher so gewählt werden, daß das darzustellende Ereignis mindestens ein Hundertstel der gesamten Periode beträgt. Wird mit mehreren Kanälen gearbeitet (Oszilloskopvorsatzgerät), so sinkt die Helligkeit schon deshalb, und die notwendige Darstellungszeit erhöht sich entsprechend der Zahl der Kanäle.

Zweckmäßigerweise stellt man auf einem Kanal konstant den Takt dar, während das zu beobachtende Signal auf dem anderen Kanal liegt. Damit erhält man auf dem Schirm des Oszilloskops immer eine Zeitreferenz und kann die Impulse zu bestimmten Teilen des Zyklus zuordnen. Zur Einstellung des Triggerpunktes sollte auch der Kurvenverlauf des Triggers relativ zum Takt dargestellt werden. Dabei ist mit dem externen Triggereingang zu arbeiten.

Grundsystem: CPU und EPROM

Im ersten Schritt soll das System soweit zur Funktion gebracht werden, daß die weiteren Untersuchungen mit Hilfe des Prozessors durchgeführt werden können. Da noch nicht sichergestellt ist, daß das RAM funktioniert, müssen die Programme auf die Benutzung von RAM-Zellen (zum Beispiel zur Speicherung von Rücksprungadressen) verzichten. Für diesen Test sollten möglichst alle Bausteine bis auf CPU, EPROM und Adreßdekodierung aus dem System herausgenommen werden, da mögliche Fehler in unbenötigten Bausteinen den Test nur unnötig komplizieren.

Zu beachten ist auch, daß keine Eingänge der CPU 'in der Luft' hängen dürfen (zum Beispiel INT, NMI, BUSRQ, WAIT).

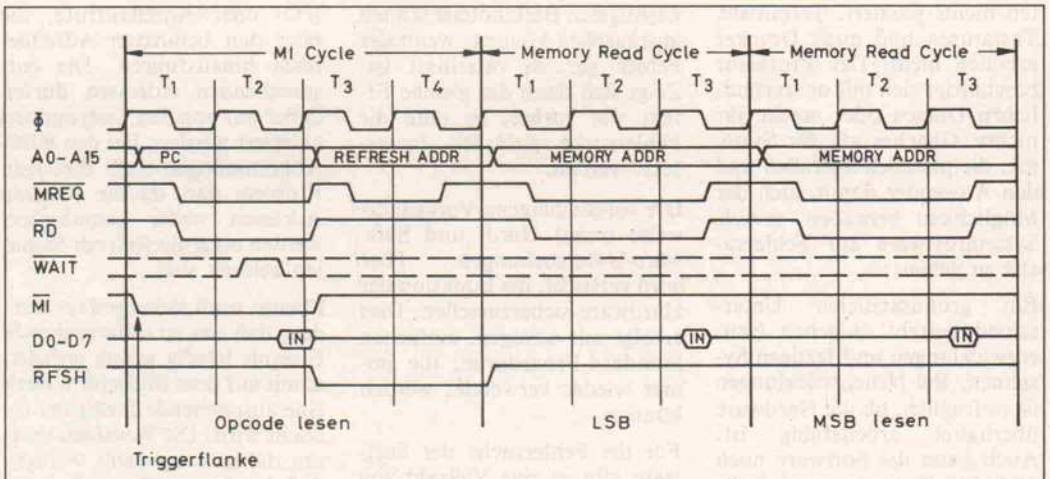


Bild 1. Impulsdiagramm zum Programm TEST 1. Gezeigt ist ein OPCODE-Ladezyklus, gefolgt von zwei RAM-Zugriffen. Getriggert wird auf die fallende Flanke von M1. Die Signale können dann auf den zwei Kanälen wiedergegeben werden.

Bei den meisten Systemen ist das unproblematisch, da diese Anschlüsse durch Logikbausteine angesteuert werden.

Für den ersten Test wird ein ganz einfaches Programm verwendet, das bei den 8080-, 8085- und Z80-Prozessoren an der Stelle 0000H in einem EPROM beginnen muß. Dieses einfache Programm (?) bietet die Möglichkeit, die Adreß-, Daten- und Steuerleitungen für RAM und ROM zu testen.

Eine endlose Schleife erzeugt eine Reihe von Signalen (Bild 1), so daß auf den genannten Leitungen für RAM und ROM mehrere Impulse pro Durchlauf erscheinen. Ein eindeutiger Triggerpunkt ist daher auf diesen Leitungen nicht zu erhalten. Benutzt man jedoch den M1-Anschluß, so erhält man eine eindeutige Triggerung, da dieser Anschluß nur einen Impuls pro Durchlauf erzeugt. Der gesamte Befehlsablauf paßt dann auf den Bildschirm. Mit Hilfe des zweiten Kanals (oder weiterer Kanäle) können die anderen Leitungen des Systems untersucht werden.

Der Ablauf des Programms kann nur durch Kurzschlüsse auf den Leitungen, eine fehlerhafte Adreßdekodierung, Leitungsunterbrechungen oder defekte Bauteile verhindert werden. Diese Fehler müssen nach alter Väter Sitte mit Ohmmeter und Überlegung beseitigt werden, denn für alle weiteren Untersuchungen ist es notwendig, daß diese Minimalkonfiguration funktioniert.

Die unteren Adreßleitungen beim Z80 lassen sich recht einfach prüfen, sie müssen sich während des Refresh-Zyklus periodisch ändern, das heißt, auf dem Bildschirm ergibt sich kein eindeutiges Bild. Die Leitungen haben zu dieser Zeit High- und Low-Pegel: tatsächlich treten diese Änderungen sehr schnell auf, so daß auf dem Schirm beide Pegel wiedergegeben werden. Die Leitungen A0...A6 dürfen während des Programms keine statischen Pegel zeigen, also nicht konstant auf 'high' oder 'low' liegen. Ist dies der Fall, liegt ein Fehler vor.

Durch die Darstellung der Datenleitungen auf dem Oszilloskop läßt sich der OPCODE (C3) eindeutig aus den Datenleitungen bestimmen. Für die anderen Adreßleitungen ist ein eigenes Programm (TEST 2) erforderlich.

Damit erzeugt man auf allen Adreßleitungen laufend Impulspakete, deren Frequenz mit zunehmender Adresse um den Faktor 2 abnimmt. Die Funktion der Adreßleitungen kann einfach geprüft werden, da die Pegel der Adreßleitungen 'high' (größer 2,0V) oder 'low' (kleiner 0,8V) sein müssen. Kurzschlüsse zwischen einzelnen Adreßleitungen erkennt man durch Zwischenpegel, die auf mindestens zwei Leitungen gleichzeitig vorhanden sind und zeitweilig deutlich von dem Pegel der anderen Adreßleitungen abweichen. Es ist allerdings zu beachten, daß derartige Pegel zulässig sind, wenn der Bus inaktiv ist, zum Beispiel während der Zyklen T3 und T4 des OP-CODE-Ladezyklus. Daher darf nur der Zeitbereich betrachtet werden, in dem die RD oder WR-Leitung aktiv ist.

Die Adreßleitungen A8...A15 können durch ihre unterschiedlichen Impulsabstände identifiziert werden. Die Abstände sind bei der Leitung A15 am niedrigsten und halbieren sich jeweils zur niedrigeren Leitung.

Mit Hilfe der Programme TEST 1 und 2 kann man sicherstellen, daß Datenbus und Adreßbus der CPU funktionieren. Daraus folgt auch, daß die Steuerleitungen MREQ, WAIT und RD in Ordnung sind.

Auch zwischen den Datenleitungen können Kurzschlüsse auftreten. Dann funktioniert vielfach schon das erste Testprogramm nicht. Es empfiehlt sich daher, die Datenleitungen mit dem Ohmmeter durchzumessen. Neben Kurzschlüssen können auch andere Bausteine Fehler, zum Beispiel Bus-Konflikte, verursachen. Vor allem I/O-Bausteine, RAMs und EPROMs kommen dabei in Betracht. Ein Bus-Konflikt entsteht, wenn die CPU Daten ausgibt, und etwa eine I/O-Einheit oder ein RAM dies zur gleichen Zeit versucht. Ursache dieses Effektes ist meist eine falsche Adressierung oder auch eine falsche Verbindung dieser Bausteine mit den RD-, WR- oder I/O-Leitungen. Derartige Effekte führen übrigens nicht immer sofort zu Fehlern.

Auch diese Fehler sind an Pegeln zu erkennen, die zwischen den sonst auf dem Bus vorhandenen Logikpegeln liegen (Bild 2). Es ist zu beachten, daß auch hier diese Zwischenpegel zulässig sind, wenn der Bus von keiner Seite bedient wird (weder RD noch WR). Liegen am Bus TTL-Bausteine, so treten in den Zeiten, in denen der Bus nicht mit Daten versorgt wird, diese Zwischenpegel, die bei ca. 2V liegen, auf. Sie werden durch die Eingangsschaltung der TTL-Schaltkreise erzeugt. Da beim Z80 der Adreßbereich und der I/O-Bereich getrennt sind, muß die Funktion der I/O-Bausteine getrennt untersucht werden.

I/O-Bausteine testen

Am Beispiel der Z80-PIO soll der Test von Ein-/Ausgabeeinheiten beschrieben werden. Wenn eine Z80-PIO (oder ein anderer I/O-Baustein) nicht funktioniert, ist in vielen Fällen eine fehlerhafte Adreßdekodierung die Ursache. In gemischten Systemen (Z80 mit 8080-Bausteinen) treten auch noch Probleme mit den Steuersignalen auf. Bei reinen Z80-Systemen beschränkt sich dies in der Regel auf Leiterbahnunterbrechungen (vergessene Leitungen), Kurzschlüsse oder Entwurfsfehler.

Der Test einer PIO ist schon etwas komplizierter, denn wenn dieser Baustein nicht initialisiert wurde, passiert nichts. Bei den folgenden Untersuchungen wird davon ausgegangen, daß die Programme TEST 1 und TEST 2 fehlerfrei ablaufen. In das Grundsystem wird nun die erste PIO eingesetzt. Aber Vorsicht: Der Eingang IEI (Pin 24) sollte auf jeden Fall mit einem 10k-Widerstand auf 'high' gezogen werden, wenn er durch eine andere PIO (Daisy Chain)

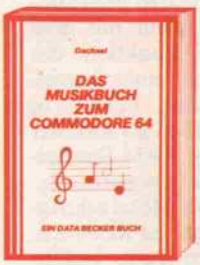
```
0000                        ORG   0H
0000 C3 00 00     MARKE     JP    MARKE     ;ENDLOSE SCHLEIFE
0000                        END   MARKE
```

TEST 1. RESET-Schleife

```
0000                        ORG   0H
0000 21 00 00     MARKE     LD    HL,0      ;Adresse vorladen
0003 7E           MARKI     LD    A,(HL)    ;Aus der Adresse lesen
0004 23                     INC   HL
0005 C3 03 00               JP    MARKI
0000                        END   MARKE
```

TEST 2. Adressen hochzählen

NEUE SUPERHITS RUND UM COMMODORE



Der COMMODORE 64 ist ein Musikgenie, und mit diesem Buch lernen Sie alles über seine musikalischen Fähigkeiten. Der Inhalt reicht von einer Einführung in die Computermusik über die Erklärung der Hardware-Grundlagen und die Programmierung in BASIC bis hin zur fortgeschrittenen Musikprogrammierung. Zahlreiche Beispielprogramme und leicht verständliche Darstellung. Erschließen Sie sich die Welt des Sounds und der Computermusik mit dem MUSIKBUCH ZUM COMMODORE 64.
Ca. 200 Seiten, DM 39,–

Graphik ist eine der Hauptstärken des COMMODORE 64. Mit diesem neuen Buch lernen Sie, wie Sie die graphischen Fähigkeiten programmtechnisch optimal nutzen, von einfachen Figuren über Sprites, Zeichensatzprogrammierung, Hardcopy und IRQ-Handhabung bis hin zu Funktionendarstellung, Statistik, 3-D, CAD und den Geheimnissen der Actionsspiele. Zahlreiche Beispielprogramme ergänzen dieses Buch, das die faszinierende Computergraphik jedermann zugänglich macht.
Ca. 250 Seiten, DM 39,–

Umfassendes Nachschlagewerk zum COMMODORE 64 und seiner Programmierung. Allgemeines Computerlexikon mit Fachwissen von A–Z und Fachwörterbuch mit Übersetzungen wichtiger englischer Fachbegriffe – das DATA BECKER LEXIKON ZUM COMMODORE 64 stellt praktisch drei Bücher in einem dar. Es enthält eine unglaubliche Vielfalt an Informationen und dient so zugleich als kompetentes Nachschlagewerk und als unentbehrliches Arbeitsmittel. Ein Muß für jeden COMMODORE 64 Anwender!
Ca. 350 Seiten, DM 49,–

Nicht nur alles über Interfaces und Ausbaumöglichkeiten des COMMODORE 64 enthält dieses Buch, sondern auch über seine vielfältigen Einsatzmöglichkeiten von der Lichtorgel über Motorsteuerung, Spannungs- und Temperaturmessung bis zur programmierbaren Stromversorgung. Zehn komplette Schaltungen zum Selberbauen, vom Eprommer über den Digital-Voltmeter mit automatischer Messbereichswahl und den Logic Analyzer bis zur preiswerten Spracheingabe-Sprachausgabe. Jeweils mit Schaltplan, Layout und Softwarelisting.
Ca. 220 Seiten, DM 49,– ab April 84.

64 FÜR EINSTEIGER ist eine sehr leicht verständliche Einführung in Handhabung, Einsatz, Ausbaumöglichkeiten und Programmierung des COMMODORE 64, die keinerlei Vorkenntnisse voraussetzt. Schritt für Schritt führt das Buch Sie in die Programmiersprache BASIC ein, wobei Sie nach und nach eine komplette Adressenverwaltung erstellen, die Sie anschließend nutzen können. Das Buch ist sowohl als Einführung als auch als Orientierung vor dem 64er Kauf gut geeignet.
Ca. 200 Seiten, DM 29,–

Diese neue, umfangreiche Programmsammlung hat es in sich. Über 50 Spitzenprogramme für den COMMODORE 64 aus den unterschiedlichsten Bereichen, vom Superspiel über Graphik- und Soundprogramme sowie Utilities bis hin zu Anwendungsprogrammen. Der Hit sind zu jedem Programm aktuelle Programmiertips und Tricks der einzelnen Autoren zum Selbermachen. Also – nicht nur abtippen, sondern dabei lernen und wichtige Anregungen für die eigene Programmierung sammeln.
Ca. 250 Seiten, DM 49,–

... und natürlich die bewährten Bestseller aus bester Hand

Insgesamt über 200 000mal wurden die nachfolgenden Bücher in nur 12 Monaten verkauft. Machen auch Sie mehr als Ihrem COMMODORE mit diesen beliebten und bewährten Bestsellern aus bester Hand.

Endlich ein umfangreiches Trainingshandbuch, das Ihnen detailliert den Umgang mit SIMON's BASIC erklärt. Ausführliche Darstellung aller Befehle und ihrer Anwendung. Zahlreiche Beispielprogramme und Programmiertricks. Dieses Buch sollte jeder SIMON's BASIC Anwender haben!
ca. 300 S., DM 49,–

Eine leicht verständliche Einführung in das Programmieren des C-64 in Maschinensprache und Assembler. Komplett mit vielen Beispielen sowie einem Assembler, Disassembler und einem Einzelschritt-Simulator. Und natürlich zugeschnitten auf Ihren Computer, den COMMODORE 64.
ca. 200 S., DM 39,–

64 INTERN erklärt detailliert Technik und Betriebssystem des C-64 und die Programmierung von Sound und Graphik. Ausführlich dokumentiertes ROM-Listing, zahlreiche lauffertige Beispielprogramme und 2 Original-Schaltpläne zum Ausklappen. Dieses Buch sollte jeder 64-Anwender und Interessent haben.
ca. 320 S.; DM 69,–

64 TIPS & TRICKS ist eine echte Fundgrube für jeden COMMODORE 64 Anwender. Umfangreiche Sammlung von POKE's und anderen nützlichen Routinen, BASIC-Erweiterungen, Graphik und Farbe für Fortgeschrittene, CP/M, Multitasking, mehr über Anschluß- und Erweiterungsmöglichkeiten und zahlreiche lauffertige Programme.
ca. 290 S.; DM 49,–

64 FÜR PROFIS zeigt, wie man erfolgreich Anwendungsprobleme in BASIC löst und verrät Erfolgsgeheimnisse der Programmierprofis. 5 komplett beschriebene, lauffertige Anwendungsprogramme (z. B. Adreßverwaltung) illustrieren den Inhalt der einzelnen Kapitel beispielhaft. Mit diesem Buch lernen Sie gute und erfolgreiche BASIC-Programmierung.
ca. 320 S., DM 49,–

DAS GROSSE FLOPPY-BUCH erklärt detailliert die Arbeit mit der Floppy VC-1541, von der sequentiellen Datenspeicherung bis zum Direktzugriff, für Anfänger, Fortgeschrittene und Profis. Ausführlich dokumentiertes DOS-Listing, zahlreiche lauffertige Beispiel- und Hilfsprogramme, z. B. Disk Editor und Haushaltsbuchführung.
ca. 320 S.; DM 49,–

VC-20 INTERN ist für jeden interessant, der sich näher mit Technik und Maschinenprogrammierung des VC-20 auseinandersetzen möchte. Detaillierte technische Beschreibung des VC-20, ausführliches ROM-Listing, Einführung in die Maschinenprogrammierung und 3 Original-Schaltpläne.
ca. 230 S.; DM 49,–

VC-20 TIPS & TRICKS ist eine echte Fundgrube für jeden VC-20 Anwender. Sound und Graphik Programmierung, Speicherbelegung und Speichererweiterung, BASIC-Erweiterungen, POKE's und andere nützliche Routinen, zahlreiche lauffertige Beispiel- und Anwendungsprogramme und vieles andere mehr.
ca. 230 S.; DM 49,–

Darauf haben Sie gewartet!

Die neue DATA WELT ist eine Computerzeitschrift speziell für COMMODORE-Anwender. Brandaktuell (detaillierte Informationen über die neuen COMMODORE Computer 264 und 364) und randvoll mit Berichten, Trends und interessanten Programmiertips. 80 Seiten stark im Magazinformat. Gleichzeitig als Nachfolger des VC-Infos umfassende Übersicht über aktuelle Produkte, Bücher und Programme rund um COMMODORE 64 und VC-20. Die Frühjahrsausgabe der neuen DATA WELT erhalten Sie ab Anfang März überall dort, wo es DATA BECKER BÜCHER und -PROGRAMME gibt. Am besten gleich holen oder direkt bei DATA BECKER gegen DM 4,– in Briefmarken anfordern.

IHR GROSSER PARTNER FÜR KLEINE COMPUTER

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010 · im Hause AUTO BECKER

DATA BECKER BÜCHER und PROGRAMME erhalten Sie im Computer-Fachhandel, in den Computerabteilungen der Kauf- und Warenhäuser und im Buchhandel. Auslieferung für Österreich Fachbuch-Center ERB, Schweiz THALI AG und Benelux COMPUTERCOLLECTIEF.

BESTELL-COUPON c't 5

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
☐ DATA WELT 1/84 (DM 4,– in Briefmarken liegen bei)

Name und Adresse bitte deutlich schreiben

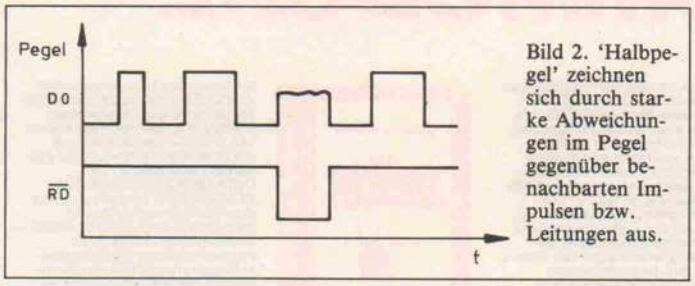


Bild 2. 'Halbpegel' zeichnen sich durch starke Abweichungen im Pegel gegenüber benachbarten Impulsen bzw. Leitungen aus.

angesteuert wird, die noch nicht eingebaut ist.

Mit dem Programm TEST 3 kann zunächst geprüft werden, ob die PIO nicht völlig falsch beschaltet ist. Da die PIO nicht initialisiert ist, sollte sie in dieser Betriebsart keine Daten ausgeben, der Datenbus wird also auch nicht auf definierte Pegel gesetzt. Bei diesem Programm wird auf IOREQ getriggert, um festzustellen, ob die PIO überhaupt adressiert wird. Bei richtiger Adressierung muß der CE-Anschluß (Pin 4) auf Low-Pegel liegen.

Für PD, das hier mit 0 angenommen wird, müssen Sie natürlich die tatsächliche Adresse des PIO-Bausteins in Ihrem System einsetzen. Die Adressen innerhalb der PIO werden durch den Anschluß PORT B/A festgelegt. CONTROL/DATA-Select entscheidet, ob Daten oder CONTROL-Kommandos an die PIO übergeben werden (eine ausführliche Beschreibung der Z80-PIO findet sich in der c't 12/83). Auf den Pins 5 und 6 muß die durch PD eingestellte Select-Kombination erscheinen. Weiterhin muß der RD-Anschluß auf Low-Pegel liegen.

Das Programm liest den Port PD und erzeugt so die entsprechenden Signale auf dem Bus (Bild 3). Dabei sind die Datenleitungen auf die schon angesprochenen Halbpegel zu untersuchen.

Zur Untersuchung der Arbeitsweise muß die PIO initialisiert werden. Dazu müssen die Adressen für die Daten und die Control-Bytes bekannt sein. Die Adressen werden hier mit PD und PCON angenommen, wobei die Selektion der PIOs im IC durch die Adreßleitungen A0 (Daten) und A1 (Control) erfolgt (dies kann bei jedem System anders gelöst sein, daher müssen die Adressen entsprechend angepaßt werden). Die Initialisierung setzt die PIO in die CONTROL-Mode, wobei alle Leitungen als Eingang eingestellt werden (TEST 4).

Auf eine Besonderheit im Programm sei noch hingewiesen. In TEST 4 wird die Control-Mode mit '0FFH' definiert. Dies ist möglich, weil die Bits 4 und 5 beliebig gewählt werden können. Die Definition durch 0FFH hat dann den Vorteil, daß die Initialisierung unabhängig von einer eventuellen Vertauschung von Busleitungen an der PIO ist.

Mit dem Oszilloskop triggert man wieder auf die fallende Flanke des IORQ-Signals. Nun kann das Signal auf dem Datenbus dargestellt werden. Stellt man bestimmte Eingangspegel an der PIO ein, beispielsweise mit einem DIL-Schalter, so müssen die entsprechenden Pegel auf den Datenleitungen wieder erscheinen. Auf diese Weise wird die Eingabe von Daten über die PIO geprüft.

Das Testprogramm 5 prüft die Ausgabe von Daten über die PIO.

Es werden auf den Ausgängen einer PIO Rechtecksignale erzeugt, wobei D0 die höchste und D7 die niedrigste Frequenz hat. Die Frequenzen unterscheiden sich jeweils um den Faktor 2. Das Register A wird vom Programm als Zähler benutzt, der jeweilige Zustand des Registers erscheint auf den Ausgangsleitungen der PIO. Durch die Wahl der Befehle zur Initialisierung erfolgt diese auch bei vertauschten Datenleitungen richtig. Die ausgegebene Frequenz erlaubt die Feststellung der Zuordnung von PIO-Leitungen zu den Datenleitungen.

```
0000                            ORG     0H
0000                    PD:     EQU     0

0000  DB  00            MARKE1: IN      A,PD            ;PIO LESEN
0002  C3  00  00                JP      MARKE1

0000                            END     MARKE1
```

TEST 3. PIO lesen (uninitialisiert)

```
0000                            ORG     0H
0000                    PD      EQU     0       ;DATENPORT
0001                    PCON    EQU     1       ;CONTROL-PORT
00FF                    CONMD   EQU     0FFH    ;CONTROL-MODE
00FF                    INPT    EQU     0FFH    ;ALLE PIN'S EINGANG
0000                    OUTP    EQU     0H      ;ALLE PIN'S AUSGANG

0000  3E  FF            INIT:   LD      A,CONMD ;PIO IN CONTROLMODE
0002  D3  01                    OUT     PCON,A
0004  3E  FF                    LD      A,INPT  ;ALLE LEITUNGEN AUF EINGANG
0006  D3  01                    OUT     PCON,A

0008  DB  00            MARK:   IN      A,PD            ;PIO LESEN
000A  C3  08  00                JP      MARK

0000                            END     INIT
```

TEST 4. PIO für Eingabe initialisieren und lesen

```
0000                            ORG     0H
0000                    PD      EQU     0       ;DATENPORT
0001                    PCON    EQU     1       ;CONTROL-PORT
00FF                    CONMD   EQU     0FFH    ;CONTROL-MODE
00FF                    INPT    EQU     0FFH    ;ALLE PIN'S EINGANG
0000                    OUTP    EQU     0H      ;ALLE PIN'S AUSGANG

0000  3E  FF            INIT:   LD      A,CONMD ;PIO IN CONTROLMODE
0002  D3  01                    OUT     PCON,A
0004  3E  00                    LD      A,OUTP  ;ALLE LEITUNGEN AUF AUSGANG
0006  D3  01                    OUT     PCON,A

0008  D3  00            MARK:   OUT     PD,A            ;AUSGEBEN AUF PIO
000A  3C                        INC     A               ;VERSCHIEDENE IMPULSE
000B  C3  08  00                JP      MARK            ;ERZEUGEN

0000                            END     INIT
```

TEST 5. PIO für Ausgabe initialisieren und schreiben

RAM-Test

Alle bisher benutzten Programme arbeiteten allein mit dem EPROM. Die Funktion des RAM kann mit einer kurzen Programmsequenz (TEST 6) untersucht werden. Es wird der CE-Eingang der RAM-Bausteine zur Triggerung benutzt. Dadurch wird eine Mehrfachtriggerung auf die EPROM-Zugriffe in jedem Falle verhindert. Im Programmbeispiel liegt das EPROM bei 0000H und das RAM bei 2000H. Durch diese Aufteilung kann auch auf die Adreßleitung A13 getriggert werden.

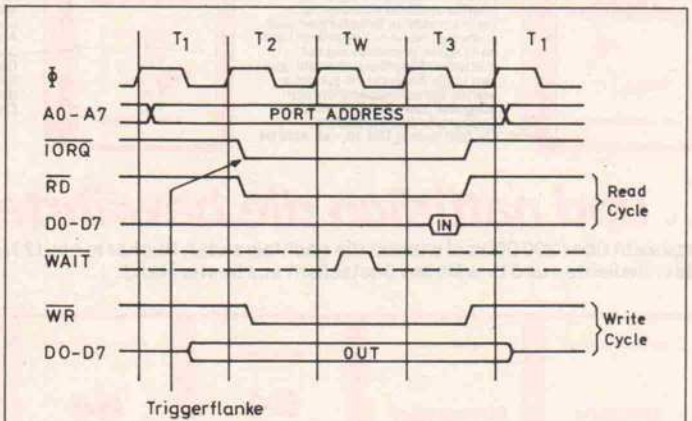


Bild 3. Impulsverläufe bei I/O-Zugriff. Die Triggerung erfolgt auf der fallenden Flanke des IORQ. Die Daten können innerhalb der nächsten 3 Taktzyklen beobachtet werden.

In jedem Falle ist die Triggerung etwas schwierig, da zwei CE- oder Adreßimpulse auftreten. Umgangen wird dieses Problem durch die Einfügung von mehreren NOPs (Leeroperationen), die eine Verlängerung des Zyklus bewirken. Wird die Ablenkzeit des Oszilloskopes kürzer als die Zykluszeit eingestellt, so wird immer auf den Schreibzugriff getriggert, und es ergibt sich ein stehendes Bild.

Bild 4 zeigt die Impulsverläufe für den Schreib- und Lesezyklus ab dem Triggerpunkt. Im ersten Teil des Bildes müssen die richtigen Daten anliegen, die im zweiten Teil wieder aus dem RAM gelesen werden. Dazwischen liegt ein Befehlszyklus, in dem der Lesebefehl aus dem EPROM geholt wird.

Mit Hilfe des Oszilloskopes ist damit sogar die Zugriffszeit des RAM bestimmbar: Es ist dies die Zeit, die im Lesezyklus vom Anlegen des MREQ bis zum Erscheinen der Daten auf dem Bus vergeht.

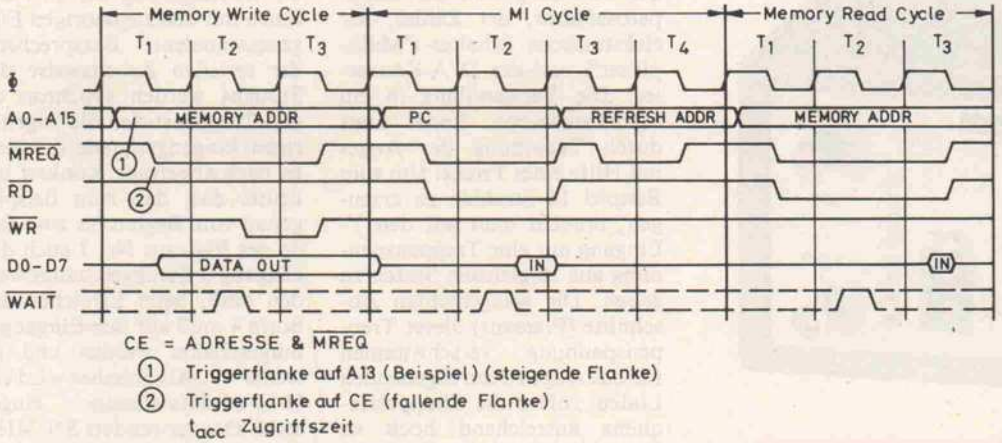


Bild 4. Impulsverläufe eines Schreib- und Lesezyklus mit zwischenliegendem M1-(Befehls-)Zyklus. Getriggert wird auf A13 oder den CE-Anschluß der RAMs.

Im Schreibzyklus eingeschriebene Daten müssen im Lesezyklus unverändert wieder aus dem RAM ausgegeben werden. Halbpegel sind bei der Ein- und Ausgabe nicht zulässig.

Einen sehr einfachen Speichertest kann man durch Beschreiben aller RAM-Zellen erreichen.

TEST 7 stellt sicher, daß die einzelnen Speicherzellen beschreibbar sind, Adreßvertauschungen fallen dabei aber nicht auf. Mit TEST 2 sind jedoch zumindest die höheren Adressen dahingehend prüfbar. Wird ein Fehler erkannt, so hängt das Programm bei der entsprechenden Speicherstelle in einer endlosen Schleife. Deren Adresse kann (mit dem Oszilloskop) an den Adreßleitungen während des Speicherzugriffs abgelesen werden. Ist das RAM einwandfrei, so läuft der Prozessor in einer kurzen Schleife im EPROM, und die höheren Adreßleitungen liegen dauernd auf Low-Pegel.

Interrupts

Bei der Verarbeitung von Interrupts treten meistens Probleme auf. Es muß vorausgesetzt werden, daß der Prozessor alle vorhergehenden Tests bestanden hat.

Die Impulsverläufe beim Interrupt werden getestet, indem ein periodischer Interrupt erzeugt wird (TEST 8). Die Programme dazu lassen sich recht einfach gestalten, da ja nur die Funktion der Hardware geprüft werden soll. Zunächst werden die PIO und der Prozessor für Interrupts initialisiert. Es folgt eine Warteschleife für den Prozessor, der damit die notwendigen Triggersignale zur Untersuchung der Interrupthardware liefert.

```
0000                    ORG     0H
2000                    ORG     2000H
0000 3E AA              LD      A,10101010B
0002 32 00 20           LD      (2000H),A
0005 3A 00 20           LD      A,(2000H)
0008 00                 NOP
0009 00                 NOP
000A 00                 NOP
000B 00                 NOP
000C 00                 NOP
000D C3 02 00           JP      0002H
0000                    END     0000H
```

TEST 6. RAM beschreiben und auslesen

```
0000                    ORG     0H              ;EPROM(!)
2000                    ORG     2000H           ;ERSTE RAM-ADRESSE
0800                    ORG     800H            ;BYTE IM RAM

0000 0E 55              LD      C,01010101B     ;TESTMUSTER
0002 21 00 20           LD      HL,2000H        ;RAMANFANG
0005 11 00 08           LD      DE,0800H        ;ZAHL DER ZELLEN
0008 79                 LD      A,C             ;TESTMUSTER LADEN
0009 77                 LD      (HL),A          ;IN RAM ABLEGEN
000A BE                 CP      (HL)            ;VERGLEICHEN
000B C2 08 00           JP      NZ,0008H        ;DANN FEHLERROUTINE
000E 2F                 CPL                     ;10101010 LADEN
000F 77                 LD      (HL),A          ;IN RAM ABLEGEN
0010 BE                 CP      (HL)            ;VERGLEICHEN
0011 C2 08 00           JP      NZ,0008H        ;DANN FEHLERROUTINE
0014 23                 INC     HL              ;NAECHSTE ADDRESSE
0015 1B                 DEC     DE              ;BYTE ZAEHLEN
0016 AF                 XOR     A
0017 BB                 CP      E               ;FERTIG?
0018 C2 08 00           JP      NZ,0008H
001B BA                 CP      D
001C C2 08 00           JP      NZ,0008H        ;WEITER!
001F C3 1F 00           JP      001FH           ;SCHLEIFE: RAM OK!
0000                    END     0000H
```

TEST 7. RAM-Test

Der Interrupt wird durch periodische Rechtecksignale am STROBE-Eingang (Pin 16 oder 17) ausgelöst. Die Taktfrequenz des Rechtecksignals sollte bei etwa 1 kHz liegen. Getriggert wird auf die steigende Flanke des HALT-Signals. Da der Prozessor im HALT-Zustand (s. TEST 8) NOPs (keine Operation) ausführt, erfolgt die Interruptlösung in jedem Falle innerhalb dieses Zyklus. Beobachtet werden können die Interruptanforderung der PIO (INT), die Signale an den PIOs (zum Beispiel IEI) und die Interruptbestätigung des Prozessors. Bild 5 zeigt den Impulsablauf einer Interruptanforderung.

```
0000                    ORG     0H
2800                    STACK   EQU     2800H   ;ENDE DES RAMBEREICHES
0000                    PD      EQU     0       ;DATENPORT
0001                    PCON    EQU     1       ;CONTROL-PORT
004F                    MODE    EQU     01001111B ;PIO-MODE(INPUT)
0087                    INTW    EQU     10000111B ;INTERRUPT FREIGABE

0000 31 00 28           INTI:   LD      SP,STACK        ;STACK FUER RUECKSPRUNG-
0003 ED 5E                      IM      2               ;ADRESSEN SETZEN
0005 3E 00                      LD      A,INTTAB/256    ;ADRESSE DER INTERRUPT-
0007 ED 47                      LD      I,A             ;TAFEL IN I-REG. LADEN
0009 3E 20                      LD      A,INTTAB        ;LSB DER TAFEL LADEN
000B D3 01                      OUT     (PCON),A        ;AN PIO SENDEN
000D 3E 4F                      LD      A,MODE          ;PIO AUF INPUTMODE
000F D3 01                      OUT     (PCON),A        ;AN PIO SENDEN
0011 3E 87                      LD      A,INTW          ;INTERRUPT ZULASSEN
0013 D3 01                      OUT     (PCON),A        ;AN PIO SENDEN
0015 FB                         EI                      ;Z80 F.INTERR. FREI GEBEN
0016 76                 ILOP:   HALT                    ;HALT-ANSCHLUß AKTIVIEREN
0017 C3 16 00                   JP      ILOP            ;SCHLEIFE

                        ;INTERRUPTROUTINE

001A FB                 INTPR:  EI                      ;INTERRUPT WIEDER FREIG.
001B ED 4D                      RETI

0020                            ORG     20H             ;INTERRUPTTAFEL
0020 1A 00              INTTAB: DW      INTPR           ;ADRESSE DER ROUTINE

0000                            END     INTI
```

TEST 8. Prüfung der Interrupt-Verarbeitung

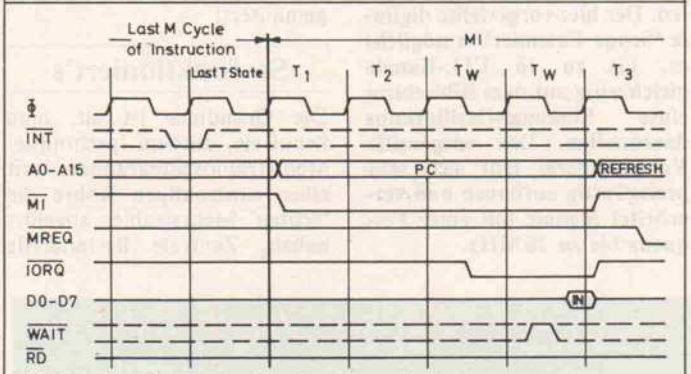
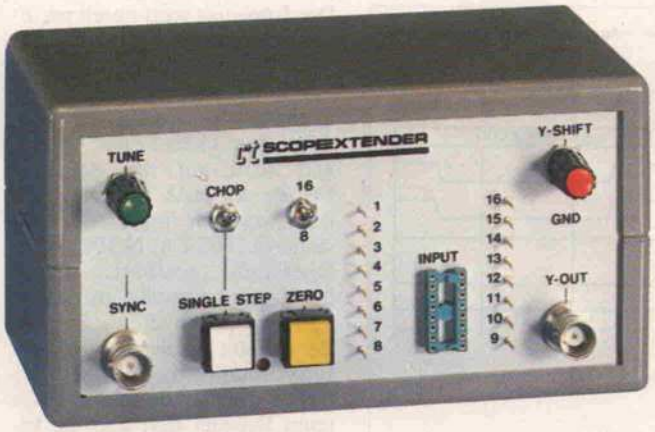


Bild 5. Impulsverläufe einer Interruptanforderung.

Mit den vorliegenden Programmen lassen sich die wichtigsten Funktionen eines Z80-Prozessors mit dem Oszilloskop testen. Erweiterungen auf andere Z80- und 8080-Bausteine sind ohne große Probleme möglich. Mit minimalem Aufwand kann damit ein Mikroprozessorsystem zum Laufen gebracht werden, ohne einige kDM in Logik-Analysatoren oder Entwicklungssysteme zu investieren. □



Scope Extender

Digitales Vorschaltgerät bringt 16 Kanäle auf den Bildschirm

H. J. Heckert

Jeder, der sich mit Computer-Hardware ernsthaft befaßt, wird sehr bald mit dem Problem des 'timings' logischer Abläufe konfrontiert. Ein Oszilloskop wird rasch zum unentbehrlichen Meßinstrument. Allerdings ist es ein sehr mühsames Verfahren, mit Hilfe eines Ein- oder Zweikanalers ein 'Timing-Diagramm' zu erstellen. Der hier vorgestellte digitale 'Scope Extender' ermöglicht es, bis zu 16 TTL-Kanäle gleichzeitig auf dem Bildschirm eines Einkanal-Oszilloskops darzustellen. Das ausgereifte Vorschaltgerät läßt sich sehr preisgünstig aufbauen und verarbeitet Signale mit einer Frequenz bis zu 20 MHz.

Da die Platine neben den Potis, Buchsen und Schaltern lediglich vier IC's, 12 Widerstände und vier Kondensatoren enthält und keinen Abgleich erfordert, sollte es auch dem weniger versierten Einsteiger möglich sein, sich solch ein Gerät selbst zusammenzubauen. Immerhin hat sich mit diesem kleinen 'Wunderding' der bescheidene Hameg 307 des Verfassers zu einem 16-Kanal-Oszilloskop gemausert!

So funktioniert's

Die Grundidee ist alt, man kennt sie, seitdem 'gechoppte' Mehrstrahloszillographen mit einer einstrahligen Röhre die 'echten' Mehrstrahler abgelöst haben. Zentrale Bestandteile der Schaltung sind der Chopperoszillator, der Zähler, der elektronische Schalter ('Multiplexer') und der D/A-Konverter. Die Verwandlung in ein Mehrkanalscope findet statt durch Täuschung des Auges mit Hilfe eines Tricks: Um zum Beispiel 16 Strahlen zu erzeugen, braucht man auf den Y-Eingang nur eine Treppenspannung aus 16 gleichen Stufen zu geben. Die waagerechten Abschnitte (Plateaus) dieser Treppenspannung verschwimmen für das Auge zu durchgehenden Linien, sofern die Chopperfrequenz ausreichend hoch ist (typ. 100 kHz) und jeder neue Strahldurchlauf an einem anderen Punkt der Treppenspannung asynchron beginnt. Letzteres ergibt sich von selbst, solange man nicht auf die Chopperfrequenz triggert.

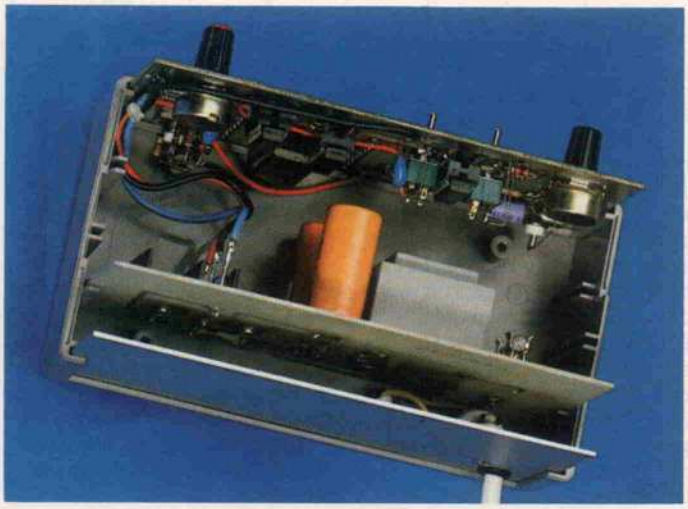
Man gewinnt dieses Treppenspannungssignal, indem man mit Hilfe des Chopperoszillators (IC4a und T1) und des Vier-Bit-Zählers (IC2) ständig der Reihe nach die Zahlen von 0 bis 15 binär als Vier-bit-Wort erzeugt. Dieses wird mit IC 3 a-d invertiert und mit den binär gewichteten Widerständen R6 bis R9 in eine analoge Treppenspannung verwandelt. Das ganze Verfahren stellt letztlich die Erzeugung eines scheinbar parallelen Strahlenbündels aus tatsächlich seriellen Teilstücken, nämlich den immer wiederkehrenden Treppenplateaus, dar.

Jetzt haben wir zwar 16 waagerechte Linien, jedoch noch keine Abbildung der 16 Logiksignale. Diese erhält man durch die Überlagerung der 16 Nullinien mit den zugehörigen Eingangssignalen. Entsprechend der seriellen Arbeitsweise des Systems werden synchron zu den Treppenstufen die zugehörigen Eingangssignale der Reihe nach abgefragt. Konkret bedeutet das, daß zum Beispiel genau vom Beginn bis zum Ende des Plateaus Nr. 3 auch der Eingang 3 durchgeschaltet werden muß, beim Erreichen der Stufe 4 muß auf den Eingang 4 umgeschaltet werden und so weiter ... Als Schalter wird ein '1-of-16-data-selector' eingesetzt. Der verwendete SN 74150 ist im Vergleich zu Analogmultiplexern wie dem HEF 4067 wesentlich schneller (t_{pd} = 11 ns) und erheblich robuster. Der Adreßeingang des Multiplexers liegt parallel auf dem vom Zähler gelieferten Vierbit-bus. Dadurch wird zu jeder Zeit der richtig zugeordnete Eingang durchgeschaltet.

Numeriert man die 16 Strahlen des Scopes von oben beginnend von 1 bis 16, so werden die Pins 1 bis 16 der Eingangsbuchse (IC-Fassung) in der richtigen Reihenfolge dargestellt.

Die serielle Abfrage nach ansteigenden Eingangsnummern bei gleichzeitig fallender Darstellung von 1 (oben) bis 16 (unten) erfordert die Invertierung des Vier-bit-Logiksignals für die Treppenspannungserzeugung (IC 3 a-d). Bild 1 zeigt die Entstehung des 16-Kanal-Bildes.

Hier noch einige Anmerkungen zum D/A-Wandler und zur Ausgangsbeschaltung: Damit



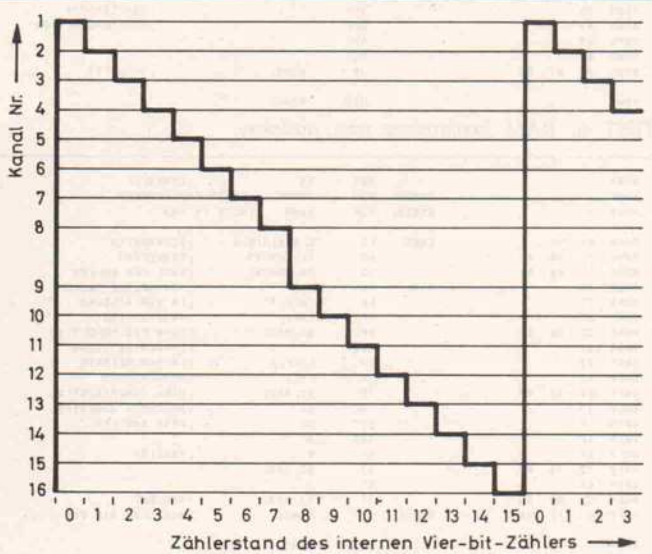


Bild 1. So entsteht das 16-Kanal-Oszillogramm

überhaupt saubere Linien zustande kommen können, muß das Treppensignal bei Chopperfrequenzen im Bereich von 100 kHz bis 1 MHz möglichst exakt sein. Einerseits sollten die Treppensprünge möglichst steil sein, weil zu langsame Flanken zu einer Aufhellung zwischen den Kanälen führen. Standard-CMOS-Logik ist deshalb bei +5 V Versorgungsspannung zu langsam. Andererseits führen 'verbogene' Treppenplateaus, wie sie von TTL-Treibern erzeugt werden, zu einem mehr oder weniger breiten 'Verschmieren' der Linien. In dieser Hinsicht kritisch ist der oberste Strahl.

Die besten Ergebnisse erzielt man mit der neuen High-Speed-CMOS-Serie (74 HC...-Familie): Diese verbinden die saubere Signalform von CMOS-Logik mit der Schnelligkeit von LS-Bausteinen. Dabei sind die Ausgänge um eine Größenordnung niederohmiger als bei CMOS, so daß erheblich höhere Ströme entnommen werden können. Das anschließende (2^n-gewichtete) Widerstandsnetzwerk aus R6 bis R9 enthält noch eine Besonderheit: Indem der Widerstand R6 absichtlich etwas zu niedrig dimensioniert wird, ergibt sich eine zu starke Gewichtung des 'most significant bit' (='MSB', hier die Zahl $2^3=8$). Dies führt zu einem vergrößerten Abstand zwischen den Treppenstufen 8 und 9 (siehe Bild 1), entsprechend den internen Zählständen 0111 = 7 beziehungsweise 1000 = 8. Auf diese Weise wird das sonst völlig gleichmäßige 16-Kanal-Diagramm zu zwei 8-Kanal-Blöcken auseinandergezogen. Dies erleichtert die Orientierung.

Auf den Sammelpunkt des D/A-Wandler-Widerstandsnetzwerkes wird mit R10 das vom Datenselektor jeweils angewählte Eingangssignal gekoppelt. Über R12 wird außerdem eine mit P2 einstellbare negative Strahlverschiebungskomponente zugemischt. Mit dem Belastungswiderstand R11 wird die Gesamtimpedanz auf 50 Ohm reell abgestimmt, so daß man diesen Ausgang direkt auf das BNC-Kabel geben kann. Andererseits ist die sich ergebende Abschwächung so ausgelegt, daß die Spannungsdifferenz zwischen zwei benachbarten Strahlen etwa 70 mV beträgt. Dies ermöglicht die optimale 16-Kanal-Darstellung beispielsweise auch auf dem 7 cm-Schirm des HM 307 bei einer Empfindlichkeit von 0,2 V/cm.

Platine und Frontplatte zugleich

Die Platine erscheint auf den ersten Blick etwas ungewöhnlich: Auf der einen Seite befinden sich die Leiterbahnen mit nicht gebohrten Lötaugen, auf der anderen zeigt sich die Frontwandbedruckung. Dahinter steckt ein einfaches Konzept: Eine Epoxydplatine ist stabile Frontwand und Leiterplatte zugleich. Diese Lösung ist äußerst kompakt, erspart die gesonderte Frontplatte und einiges an Verdrahtung.

Der Zusammenbau ist kinderleicht, wenn man sich an die angegebene Reihenfolge hält: Zunächst werden die Löcher aufgebohrt, und zwar BNC-Buchsen auf 9,5 mm, Potis je nach Typ auf 7 oder 10 mm, die Minischalter auf 5 mm und die LED auf 3 mm. Von der Frontseite aus werden die 16 + 2 Lötnägel eingesetzt, mit dem Hammer eingeschlagen und auf der Rückseite angelötet. Als Unterlage empfiehlt sich hierfür ein Brett auf weichem Holz, keinesfalls Preßspan.

Die Eingangsbuchse (16polige IC-Fassung) wird ebenfalls von vorn eingesetzt (die Markierung nach oben) und auf der Rückseite sauber verlötet. Jetzt können die BNC-Buchsen montiert werden. Um eine zuverlässige HF-Erdung sicherzustellen, sollte man die Befestigungsmuttern mit der masseführenden Leiterbahnfläche verlöten. Die beiden Potentiometer werden so eingesetzt, daß bei entsprechend herabgebogenen Anschlußfahnen diese mit den entsprechenden Leiterbahnen verlötet werden können. Gut geeignet sind die 4-mm-Ausführungen von Radiohm; bei den meisten anderen Herstellern müssen die Kontakte über zusätzliche kurze Drahtbrücken hergestellt werden.

Nun kann man mit der Bestückung der restlichen passiven und aktiven Bauteile beginnen. Widerstände, Kondensatoren und ICs werden mit entsprechend gekürzten Anschlußdrähten auf die zugehörigen, ungebohrten Leiterpunkte gelötet (siehe Bestückungsplan). Ängstliche Naturen können auch IC-Fassungen verwenden, in jedem Falle ist auf die richtige Lage der ICs zu achten (sämtliche Markierungen weisen nach unten).

Die Widerstände R6 bis R12 werden jeweils direkt auf den Kontakt der BNC-Buchse gelötet, dadurch ist ein kapazitätsarmer, störungssicherer Aufbau gewährleistet. Es folgt die Bestückung der beiden Tipptasten und Mikroschalter, die LED wird von hinten durch das 3-mm-Loch gesteckt und die abgewinkelten Zuleitungen auf die zugehörigen Kontaktflächen gelötet. Hierbei ist natürlich auf die richtige Polung zu achten. Schließlich werden die Schalter S1 und S2 sowie die 'Sync'-Buchse mit dünner Litze verdrahtet und die Verbindung zum Netzteil hergestellt. Nun ist der Aufbau beendet, und der Probelauf kann beginnen.

Der Scope Extender benötigt zwei gut stabilisierte Versorgungsspannungen (+5 V, 100 mA, und −5 V, 30 mA), die man im allgemeinen einem Computer-Netzteil entnehmen kann. Wer das auf unseren Fotos abgebildete kompakte Gerät aufbauen möchte, sollte sich an unseren Netzteil-Vorschlag halten. Das Gehäuse stammt aus der Verobox-Serie (Bestellnummer 075-01239K). Bitte achten Sie beim Einbau des Netzteils auf elektrische Sicherheit!

Ausprobieren

Nachdem der Netzstecker Verbindung mit 220 V aufgenommen hat, muß die Betriebsanzeige-LED aufleuchten. In jedem Falle sollte man die Betriebsspannungen überprüfen (+5 V und −5 V, jeweils ±0,5 V). Nun verbindet man über ein BNC-Kabel den 'Y-out' mit dem Eingang des Scopes (DC-Kopplung!) und stellt auf 0,1 bis 0,2 V/cm ein. Die Zeitablenkung sollte 'freilaufen'; das heißt in den mei-

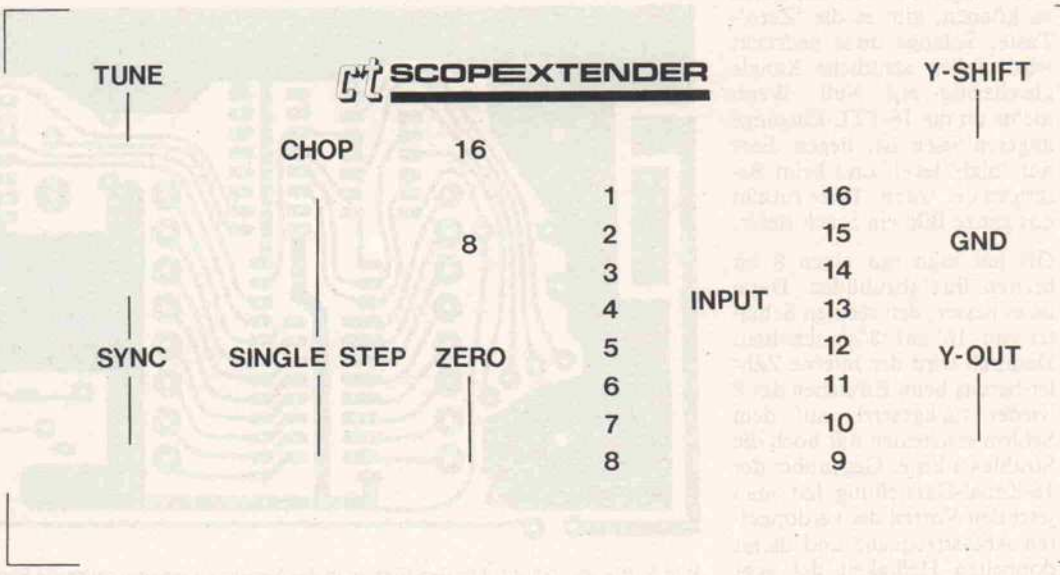


Bild 2. Vorschlag für die Frontplattenbeschriftung

sten Fällen, 'trig source' auf 'ext' und den 'trig. level' so einstellen, daß der Strahl auch ohne angeschlossenes Eingangssignal erscheint.

Wenn die beiden Schalter am Scope Extender nach oben gekippt sind und das Poti 'Y-shift' etwa in der Mittelstellung steht, sollte sich auf dem Bildschirm das 16-Kanal-Diagramm zeigen. Hierbei wird man die Intensität am Scope etwas nachregulieren müssen, denn die Gesamthelligkeit des Elektronenstrahls wird jetzt auf 16 Strahlen verteilt. Wenn das soweit funktioniert, kann man darangehen, sich mit den einzelnen Bedienungselementen vertraut zu machen.

Mit dem Poti 'Y-shift' läßt sich jeder Strahl in die Bildmitte bringen. Erhöht man nun die Y-Verstärkung auf zum Beispiel 20 mV/cm, dann lassen sich sämtliche Kanäle 'unter der Lupe betrachten', man braucht sie nur der Reihe nach durch das Bild zu schieben.

Da hierbei meistens die Helligkeit doch schon deutlich abnimmt, wurde alternativ die Einzelkanaldarstellung vorgesehen. Wenn man mit dem linken Kippschalter den Chopper abschaltet, kann man mit der darunter liegenden Tipptaste ('single step') der Reihe nach Kanal 1 bis 16 anwählen. Jetzt arbeitet der Multiplexer als statischer Schalter, der jeweils durchgeschaltete Kanal wird nicht zerhackt, und man hat wieder die volle Helligkeit.

Um auch bei statischen Signalen eindeutig 'low' beziehungsweise 'high' im Bild zuordnen zu können, gibt es die 'Zero'-Taste. Solange diese gedrückt wird, fallen sämtliche Kanäle gleichzeitig auf Null. Wenn nichts an die 16-TTL-Eingänge angeschlossen ist, liegen diese auf 'high'-level, und beim Betätigen der 'Zero'-Taste rutscht das ganze Bild ein Stück tiefer.

Oft hat man nur einen 8 bit breiten Bus abzubilden. Dann ist es besser, den rechten Schalter von '16' auf '8' zu schalten. Dadurch wird der interne Zähler bereits beim Erreichen der 8 wieder rückgesetzt, auf dem Schirm erscheinen nur noch die Strahlen 1 bis 8. Gegenüber der 16-Kanal-Darstellung hat man jetzt den Vorteil der verdoppelten Abtastfrequenz und damit doppelten Helligkeit der acht Strahlen.

Um über den gesamten Zeitablenkbereich des Scopes 'zerhackte' Bilder zu vermeiden, läßt sich die Chopperfrequenz mit dem Poti 'Tune' im Bereich von 100 kHz bis 1 MHz stufen-

Stückliste Scope Extender

(ohne Netzteil)

Widerstände (Kohleschicht, ¼ W):

R1	47R
R2	33k
R3	1k0
R4	330R
R5	330R
R6	3k9
R7	2k
R8	910R
R9	390R
R10	8k2
R11	82R
R12	270R

Potentiometer

P1	10k lin Poti mit 4-mm-Achse
P2	220R lin Poti mit 4-mm-Achse

Kondensatoren

C1	4µ7/63 V Elko axial
C2	680p Folie FKC2
C3,C4	10µ/16 V Tantalperle

Halbleiter

IC1	SN74150 TTL
IC2	SN74LS161
IC3	74HC00
IC4	74LS132
Q1	BC 547 B/ BC 237 o. ä.
LD	3-mm-LED

Verschiedenes

S1	1 Mikrokippschalter
S2	1 Mikrokippschalter
S3	1 Digitast 1x ein
S4	1 Digitast 1x ein
1	IC-Fassung, 16pol.
18	Lötnägel
2	BNC-Buchsen
2	Spannzangenknöpfe, 4 mm
1	Platine

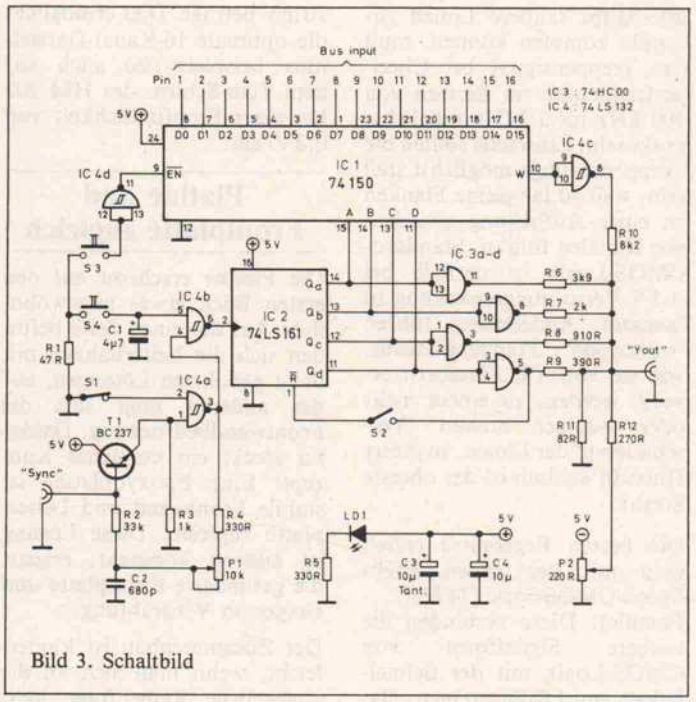


Bild 3. Schaltbild

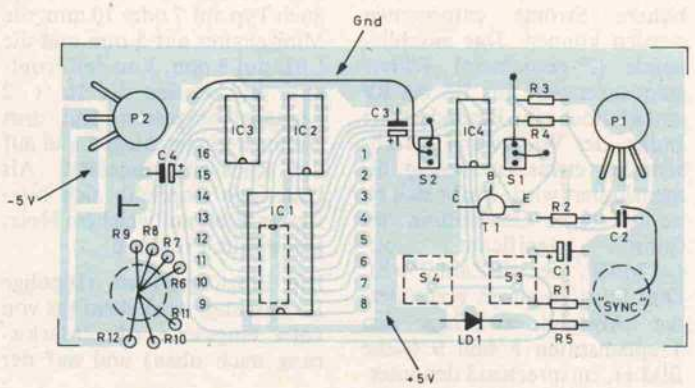


Bild 4. Bestückungsplan. Die Bauteile werden auf der Leiterbahnseite aufgelötet.

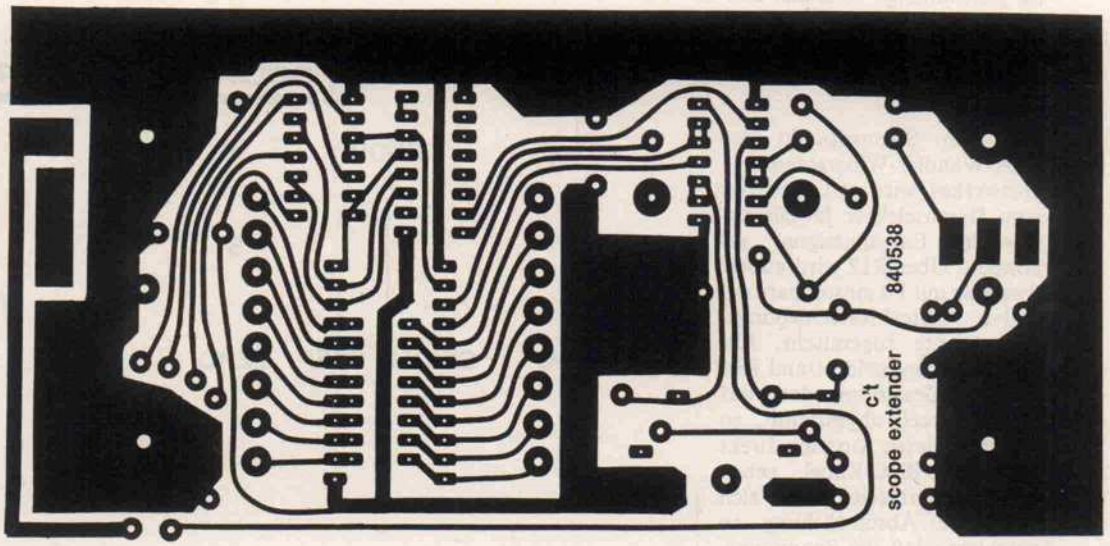


Bild 5. Platinenlayout. Für den Einbau in das Verobox-Gehäuse sollte die Platine auf das Maß 72,4 x 148 mm zugeschnitten werden.

los einstellen. Über die BNC-Buchse ('Sync') kann die Chopperfrequenz extern 'aufgezwungen' werden (TTL-Pegel); auf diese Weise ist auch ein synchron alternierender Betrieb möglich. Meistens läßt sich aber mit dem normalen Chopper auf der tiefsten Frequenz einwandfrei arbeiten.

Jetzt muß man nur noch die GND-Verbindung zum Meßobjekt herstellen und die Eingänge mit dem zu messenden Bus verbinden. Dies geschieht entweder mit Kroko-Prüfstrippen über die 16 Lötnägel oder mit einem Dil-16-Adapter über die Eingangsbuchse. Hier sind die lötfreien Quetschstecker in Verbindung mit entsprechenden Flachkabeln sehr praktisch. Der Verfasser hat sich auf dieser Basis einen einfach herzustellenden IC-Prüfadapter gebaut: An die beiden Enden eines 16poligen Fachkabels wird je ein Stecker gepreßt. Danach entfernt man vorsichtig wieder das Oberteil eines Steckers und lötet auf die herausstehenden Kontakte eine Dil-16-IC-Fassung. Jetzt braucht man nur noch das entsprechende IC aus seiner Fassung ziehen, setzt den Adapter dafür ein und steckt das IC in die Fassung auf dem Adapter ('Piggy pack'): Fertig ist die 16-fach-Verbindung vom IC zum Scope.

Wie so etwas in der Praxis aussieht, zeigt das Foto: Hier wurden Pin 1 bis 16 eines 64-k-DRAM, Typ TMS 4164-N15 'in action' dargestellt. Das Scope wurde auf dem Zyklusbeginn extern getriggert (negative Flanke von $\overline{RAS}$). Man sieht zum Beispiel, daß die Umschaltung von 'Row'- auf 'Col'-Adresse etwa auf der Mitte zwischen den fallenden Flanken von $\overline{RAS}$ (4) und $\overline{CAS}$ (15) stattfindet, data input (2) ist in jedem Falle stabil, wenn $\overline{WE}$ (3) aktiv wird und data out (14) folgt mit knapp 100 ns Verzögerung $\overline{CAS}$ (15). Weiterhin erkennt man, daß die während des Memory-Zyklus auf dem Adreßbus liegenden 'Col'-Adressen größtenteils statisch 'low' sind. Dies zeigt, daß nur ein kleiner Bruchteil des verfügbaren Speicherraumes aktiviert wurde.

Da keinerlei Speicherung vorgesehen ist, können nur periodische Vorgänge, jedoch keine Einzelereignisse aufgezeichnet werden. Dies bleibt wohl bis auf weiteres den weitaus aufwendigeren Logikanalysatoren vorbehalten. Gegenüber Speicherkonzepten hat man durch den Fortfall der 'sampling'-Technik (Signalabtastung in einem festen Zeitraster von wenigstens 10 ns) den Vorteil der Echtzeitdarstellung und damit auch die volle Ausnutzung der Low-power-Schottky-Übertragungsgeschwindigkeit. Dies führt zu einer Zeitauflösung von wenigen Nanosekunden und ermöglicht auch Messungen an sehr zeitkritischen Systemen.

Ursprünglich war dieser Scope-Extender nur für den privaten Gebrauch mit dem HM 307 des Verfassers konzipiert worden. Zusätzliche Praxistests mit Oszilloskopen der gehobeneren Preisklassen wie dem PM 3212 (25 MHz) und dem Tek 465 (100 MHz) ergaben neben einem helleren Bild eine deutliche Verbesserung der Anstiegszeiten. Dabei hat sich herausgestellt, daß bei allen Oszillographen bis einschließlich 30 MHz Grenzfrequenz die volle Bandbreite ausgenutzt werden kann.

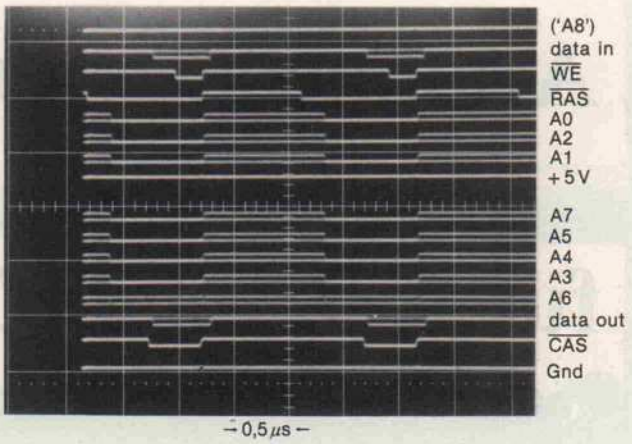


Bild 6. 64-KBit-DRAM 'in action'

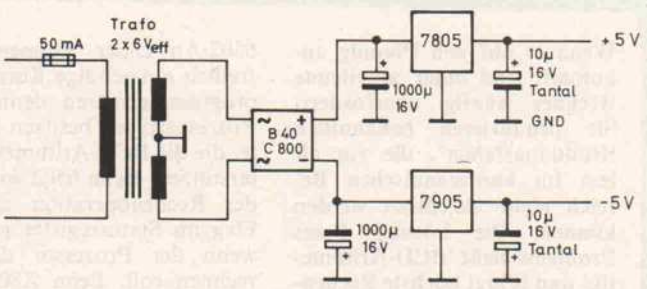
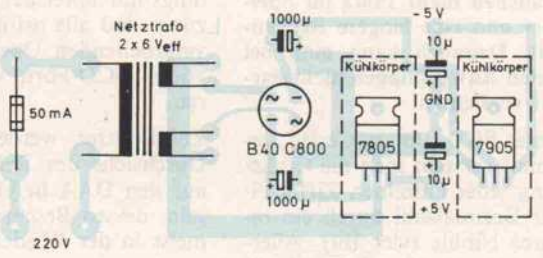
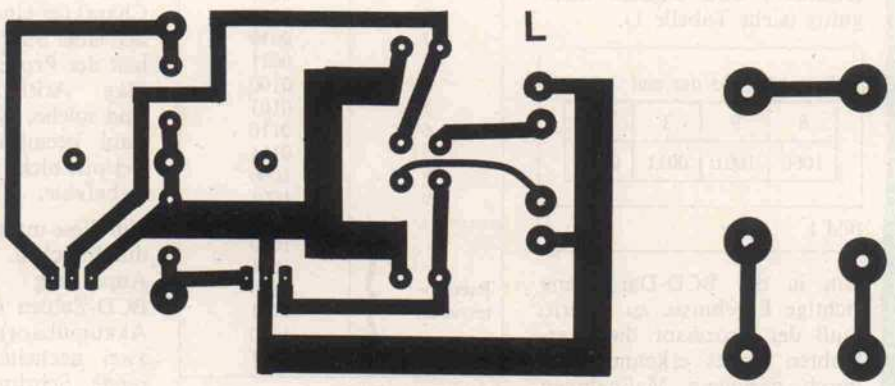


Bild 7. Vorschlag für ein geeignetes Netzteil

Stückliste Netzteil

Netztrafo 2x6 V_{eff}, 3 VA
2 Elkos 1000µF, 16-V
Gleichrichterbrücke B40 C800 o.ä.
1 7805
1 7905
2 Kühlkörper
2x 10µF, 16-V-Tantal
2 Sicherungskontakte
Sicherung 50-mA
Platine
Netzkabel u. -stecker





ASSEMBLER

Arithmetik-Unterricht

für 6502 und Z80

Ulrich Schinke

Teil 3: BCD-Arithmetik

Wenn es auf den Pfennig ankommt, sind binär arbeitende Rechner häufig überfordert. Sie produzieren bekanntlich 'Rundungsfehler', die vor allem im kaufmännischen Bereich nicht akzeptiert werden können. Die Lösung dieses Problems heißt 'BCD-Arithmetik' und bringt höchste Rechengenauigkeit bis zur letzten Stelle. Wie immer, gibt es auch eine Kehrseite der Medaille: Die Programme sind aufwendiger, brauchen mehr Platz im Speicher und eine längere Rechenzeit. Dennoch kann man bei vielen Anwendungen nicht darauf verzichten.

Unter BCD (Binary Coded Decimal) versteht man die Codierung jeder einzelnen Ziffer einer Dezimalzahl durch ein binäres Nibble (vier Bit). Allerdings lassen sich mit vier Bit 16 verschiedene Bitmuster erstellen, denen nur die zehn Ziffern 0 . . . 9 gegenüberstehen. Sechs Binärkombinationen ('Pseudotetraden') sind deshalb nicht gültig (siehe Tabelle 1).

Beispiel: 8935 dezimal

8	9	3	5
1000	1001	0011	0101

Bild 1

Um in der BCD-Darstellung richtige Ergebnisse zu liefern, muß der Prozessor die unerlaubten Codes erkennen und durch geeignete Maßnahmen korrigieren. Den prinzipiellen Ablauf zeigt Bild 2. Z80- und 6502-Anwender können sich freilich aufwendige Korrekturprogramme sparen, denn beide Prozessortypen besitzen Befehle, die die BCD-Arithmetik unterstützen. Beim 6502 wird vor der Rechenoperation das D-Flag im Statusregister gesetzt, wenn der Prozessor dezimal rechnen soll. Beim Z80 kann man nach der Rechenoperation mit dem Befehl 'DAA' den Dezimal-Abgleich ausführen lassen. Dies funktioniert allerdings nur unter der Voraussetzung, daß alle arithmetisch zu verarbeitenden Operanden zuvor in BCD-Form codiert waren.

Z80-Besitzer werden bei der Durchsicht des Befehlsvorrats auf den DAA-Befehl gestoßen sein, dessen Beschreibung sich meist in der Wiedergabe einer Tabelle erschöpft. Dieser mächtige Abgleichbefehl addiert eine Hex-Zahl zum Inhalt des Akkumulators, in Abhängigkeit von der zuvor durchgeführten arithmetischen Operation und dem Carry-Flag. Die Information, ob die zuletzt ausgeführte Operation dem Charakter einer Addition oder den einer Subtraktion hatte, erhält der Prozessor über das N-Flag. Arithmetische Befehle sind solche, die den Wert einer Zahl beeinflussen, also zum Beispiel nicht die reinen Transferbefehle.

Dezimalzahl	tetradischer Dualzahlencode 8421*
Ø	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
Pseudotetraden	1010
	1011
	1100
	1101
	1110
	1111

* Stellenwertigkeit

Tabelle 1: 'BCD'-Code

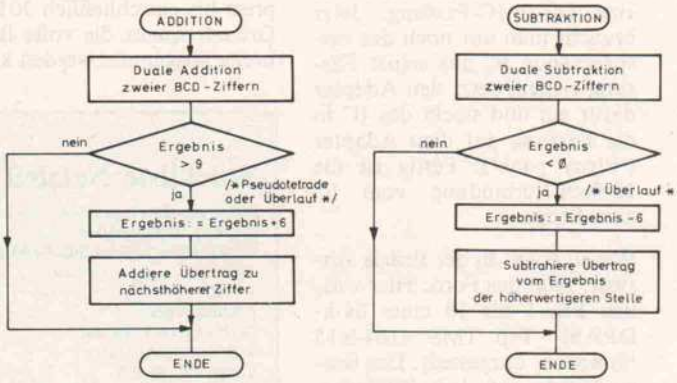


Bild 2. Dezimaler Abgleich

Um diese mysteriöse Tabelle zu durchblicken, sollte man die Anpassung der gepackten BCD-Zahlen (zwei Ziffern im Akkumulator) gedanklich in zwei nacheinander auszuführende Schritte zerlegen. Zunächst wird die niederwertige, dann die höherwertige Ziffer unter Berücksichtigung hierbei enstehender Überträge abgeglichen.

Der Z80 besitzt gegenüber dem 6502 zwei dezimale Rotationsbefehle (RLD, RRD). Sie rotieren den Inhalt der durch das HL-Register bestimmten Adresse um jeweils vier Bit, wobei das untere Akkumulatornibble dieselbe Aufgabe übernimmt, wie das Carry-Flag bei den 1-Bit-Rotier-Befehlen (Bild 3).

Dezimale Algorithmen

Die BCD-Arithmetik verlangt, daß die Operanden ziffernweise verarbeitet werden. Demgegenüber beruhen die für Binärarithmetik erstellten Algorithmen auf einer Bit-für-Bit-Behandlung der numerischen Da-

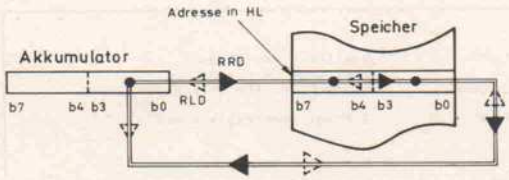


Bild 3. Dezimale Rotation beim Z 80

ten. Sie eignen sich deshalb nicht für BCD-Berechnungen.

Wie aus der Binärarithmetik bekannt, existiert eine spezielle Codierung, die Komplementdarstellung, die es ermöglicht, sowohl positive als auch negative Zahlen nach der gleichen Rechenvorschrift zu verarbeiten.

Grundsätzlich läßt sich in jedem Zahlensystem bei festgelegter Stellenzahl das Komplement bilden. Das Komplement (X') einer Integerzahl (X) mit m Stellen wird durch Subtraktion dieser Zahl von der kleinsten mit (m + 1) Stellen darstellbaren Zahl bestimmt. Mathematisch heißt dies:

$X' = b^m - X$

wobei b die Basis des gewählten Zahlensystems ist.

Beispiel:

Zehnerkompl. (71 dez.) = $10^2 - 71 = 29$ dez.

Die hier abgedruckten Programme zur Implementierung der vier Grundrechenarten erwarten in Anlehnung an kaufmännische Berechnungen, daß die Operanden als Festkommazahlen entsprechend dem Format in Bild 4 übergeben werden.

Demnach stehen dem Benutzer bei einer vor der Assemblierung festzulegenden dezimalen Wortlänge (DWL) von beispielsweise acht Bytes zur Zahlendarstellung elf Vor- und zwei Nachkommastellen zur Verfügung. Zusätzlich codieren die drei höchstwertigsten Ziffern das Vorzeichen: 000 für positive, 999 für negative, zehnerkomplementierte Zahlen. Das erste Byte dient dabei als Puffer bei der Multiplikation beziehungsweise Division, wodurch sich der Programmieraufwand erheblich vereinfacht.

Bei der Multiplikation zweier derart formatierter Dezimalzahlen ergibt sich zwangsläufig eine Zahl mit vier Nachkommastellen (Beispiel: 13,25 * 07,13 = 94,4725), bei der Division eine ganze Zahl mit Rest (Beispiel: 34,58/7,12 = 4 Rest 6,10). Um trotzdem auch die Ergebnisse im vorgegebenen Format zu erhalten, werden Schiebeoperationen zur Komma-Anpassung durchgeführt. Bei der Produktbildung entfallen somit die dritte und vierte Nachkommastelle. Der Anwender könnte an dieser Stelle ein Unterprogramm aufrufen, das die zweite Nachkommastelle rundet. Der nach einer Division im Dividendenspeicher stehende Rest ist demgegenüber mit dem Faktor 100 multipliziert:

< 34,58/7,12 = 4,85 Rest 0,048 >

00034,58 <—— Dividend
00007,12 <—— Divisor
00004,85 <—— Quotient
00004,80 <—— Rest * 100

Resumee:

Während die Addition positive wie negative zehnerkomplementierte Zahlen verarbeitet, setzen die Multiplikation und Division positive Festkommazahlen voraus. Falls dabei auch die zehnerkomplementierten, negativen Zahlen Verwendung finden sollen, müssen die Berechnungen betragsmäßig ausgeführt werden. Mit Hilfe der Komplementierungsprogramme werden die Beträge der negativen Zahlen bestimmt. Das resultierende Vorzeichen ergibt sich dann aus einer Exklusiv-oder-Verknüpfung der Operandenvorzeichen. Der Betrag eines eventuell negativen Ergebnisses muß zu guter Letzt noch einmal zehnerkomplementiert werden. Der Anwender hat dafür Sorge zu tragen, daß die Ergebnisse im gewählten Zahlenformat darstellbar sind, also kein Überlauf auftritt.

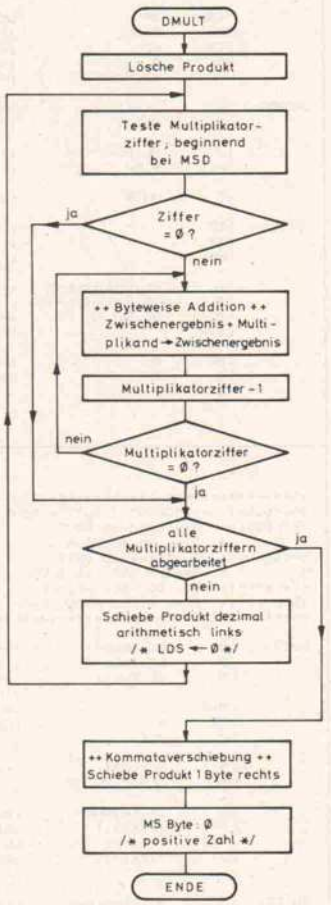


Bild 5. Ablaufplan der Multiplikation

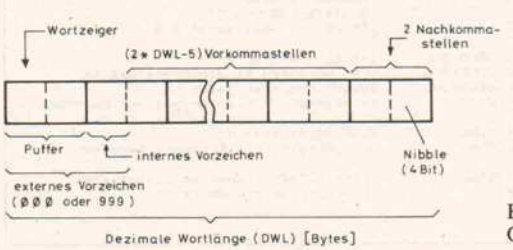


Bild 4. Format der Operanden

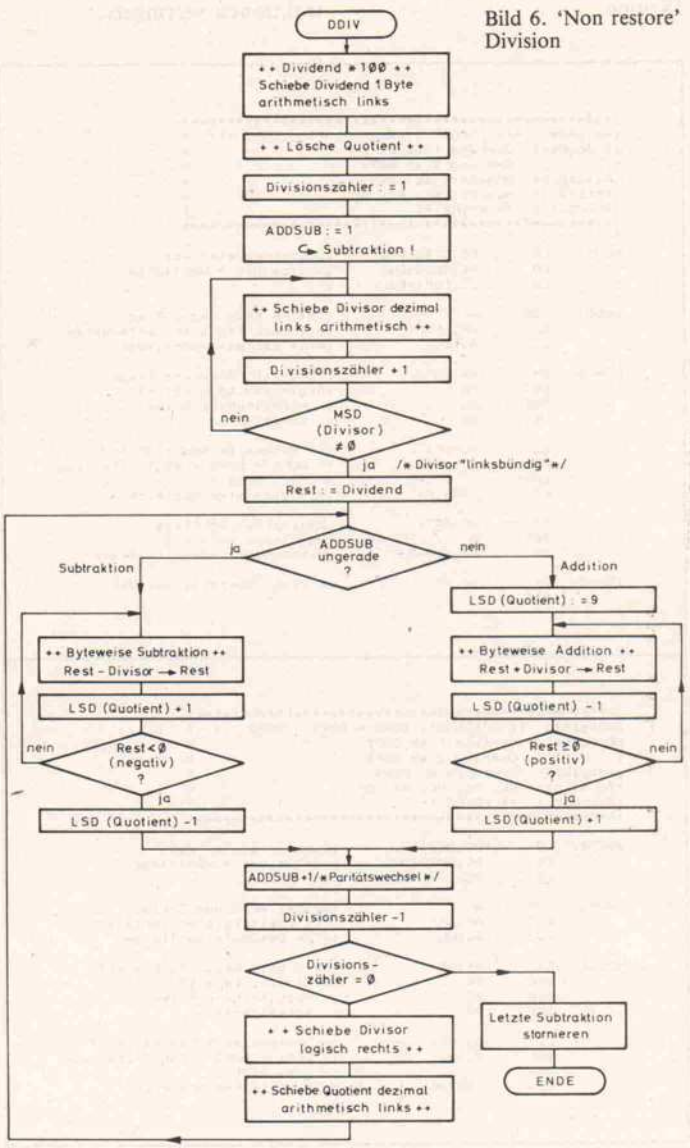


Bild 6. 'Non restore' Division

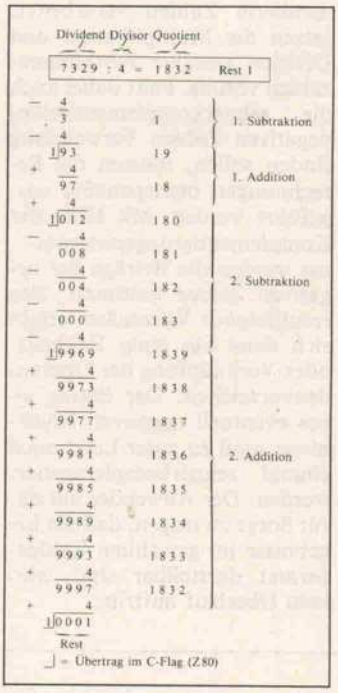


Bild 7: Beispiel zur 'Non-restore' Division

Nun noch einige Bemerkungen zu den Programmen:

Beim 6502 und Z80 wird der in Bild 5 dargestellte Multiplikationsalgorithmus unterschiedlich implementiert. Beim Z80 bietet es sich an, die höchstwertige Multiplikatorziffer links herauszuschieben. Beim 6502 wird der Multiplikator im Speicher nicht verändert, sondern Byte für Byte zunächst das höher-, dann das niederwertige Nibble getestet.

Bei dem in Bild 6 aufgezeigten Divisionsalgorithmus wechseln Additionen und Subtraktionen jeweils bei einer Vorzeichenänderung des Rests. Das in Bild 7 durchgerechnete Beispiel mag diesen etwas ungewöhnlichen Vorgang erläutern. Während der Quotientenbestimmung sind gegenüber der 'normalen Restore-Methode' keine 'Stornieroperationen' notwendig. Dadurch wird letztlich die Anzahl der Additionen und Subtraktionen verringert.

```
;*********************************************
;Aufgabe:  +++ "ADDI" : DERG = DOP1 + DOP2 +++  *
;Eingaben:  Operand 1 ab DOP1                   *
;        :  Operand 2 ab DOP2                   *
;Ausgaben:  Ergebnis ab DERG                    *
;Zerstört:  HL, DE, BC, AF, AF'                 *
;Subrout.:  ++ KEINE ++                         *
;*********************************************

ADDI:   LD      DE,DERG+DWL     ;Doppelregister auf
        LD      HL,DOP2+DWL     ;Wortzeiger + Wortlänge
        LD      BC,DOP1+DWL     ;

DADD:   XOR     A               ;Lösche Akku und C-Flag
        EX      AF,AF'          ;mit Zweitregister vertauschen
        LD      A,DWL           ;A':= Dezimale Wortlänge

NDADD:  EX      AF,AF'          ;"Orginal" Akku und Flags !
        DEC     HL              ;beginnend beim LB -->
        DEC     BC              ;  nächsthöhere Bytes
        DEC     DE              ;  adressieren

        LD      A,(BC)          ;++ Byteweise Addition ++
        ADC     A,(HL)          ;( DOP1 + DOP2 + mögl. Übertrag
        DAA                     ;  ---> DERG )
        LD      (DE),A          ;mit dezimalem Abgleich !

        EX      AF,AF'          ;Zweitakku und Flags !
        DEC     A               ;Wortlänge (A') - 1
        JR      NZ,NDADD        ;Wiederhole, für alle Bytes

OKADD:  EX      AF,AF'          ;C-Flag: Übertrag aus MSD
        RET
```

```
;*********************************************
;Aufgabe:  +++ "SUBTR": DERG = DOP1 - DOP2 +++  *
;Eingaben:  Operand 1 ab DOP1                   *
;        :  Operand 2 ab DOP2                   *
;Ausgaben:  Ergebnis ab DERG                    *
;Zerstört:  HL, DE, BC, AF, AF'                 *
;Subrout.:  ++ KEINE ++                         *
;*********************************************

SUBTR:  LD      DE,DERG+DWL     ;Doppelregister auf
        LD      HL,DOP2+DWL     ;Wortzeiger + Wortlänge
        LD      BC,DOP1+DWL     ;

DSUB:   XOR     A               ;Lösche Akku und C-Flag
        EX      AF,AF'          ;mit Zweitregister vertauschen
        LD      A,DWL           ;A':= Dezimale Wortlänge

NDSUB:  EX      AF,AF'          ;"Orginal" Akku und Flags !
        DEC     HL              ;beginnend beim LB -->
        DEC     BC              ;  nächsthöhere Bytes
        DEC     DE              ;  adressieren

        LD      A,(BC)          ;++ Byteweise Subtraktion ++
        SBC     A,(HL)          ;( DOP1 - DOP2 - mögl. Übertrag
        DAA                     ;  ---> DERG )
        LD      (DE),A          ;mit dezimalem Abgleich !

        EX      AF,AF'          ;Zweitakku und Flags !
        DEC     A               ;Wortlänge (A') - 1
        JR      NZ,NDSUB        ;Wiederhole, für alle Bytes

OKSUB:  EX      AF,AF'          ;C-Flag: Übertrag aus MSD
        RET
```

```
;*********************************************
;Aufgabe:  +++ "DMULT" : DPRD = DMTD * DMTR ++  *
;Eingaben:  Multiplikand ab DMTD                *
;        :  Multiplikator ab DMTR               *
;Ausgaben:  Produkt ab DPRD                     *
;        :  DMTR = 0                            *
;Zerstört:  HL, DE, BC, AF, AF'                 *
;Subrout.:  CLR, RLDW, DADD                     *
;*********************************************

DMULT:  LD      HL,DPRD         ;++ Lösche Produkt ++
        CALL    CLR             ;

        LD      IX,INTVAR       ;Basisadressse für interne Variablen
        LD      (IX+2),DWL*2    ;Nibblezähler:= DWL * 2

                                ;*** Nibbleweise Multiplikation ***
NNIB:   LD      HL,DMTR+DWL     ;++ Schiebe Multiplikator dezimal
        CALL    RLDW            ;  logisch links ( MSD --> A ) ++
        JR      Z,NOADD         ;Falls A=0, keine Addition !

RADD:   LD      DE,DPRD+DWL     ;++ Byteweise Addition ++
        LD      BC,DPRD+DWL     ;(Zwischenergebnis(BC) + Multiplikand(HL
        LD      HL,DMTD+DWL     ;  --> Zwischenergebnis(DE) )

        PUSH    AF              ;Multiplikatorziffer --> Stack
        CALL    DADD            ;Führe Addition aus
        POP     AF              ;Multiplikatorzifferwieder vom Stack
        DEC     A               ;holen und dekrementieren
        JR      NZ,RADD         ;Wiederhole, bis Multiplikatorziffer=0

NOADD:  DEC     (IX+2)          ;Nibblezähler - 1
        JR      Z,OK2           ;Fertig, falls Nibblezähler = 0

        LD      HL,DPRD+DWL     ;++ Schiebe Produkt dezimal
        CALL    RLDW            ;  arithmetisch links ++

        JR      NNIB            ;Verweige zu nächster Multiplikation

OK2:    NOP                     ; **** Platz für Aufruf einer
        NOP                     ; eigenen Rundungsroutine für die
        NOP                     ; 2. Nachkommastelle      ****

        LD      HL,DPRD+DWL-2   ;++ Schiebe Produkt 1 Byte rechts ++
        LD      DE,DPRD+DWL-1   ; (HL) --> (DE)=(HL)-1
        LD      BC,DWL-1        ;HL-1, DE-1, BC-1
        LDDR                    ;Wiederhole, bis BC=0

        INC     HL              ;MSB(Produkt) adressieren
        LD      (HL),0          ;MSByte :=0, da nur positive Zahlen
        RET
```

```
;*********************************************
;Aufgabe:  ++ "DDIV" : DQTT = DDVD / DDVS ++    *
;Eingaben:  Dividend ab DDVD                    *
;        :  Divisor ab DDVS                     *
;Ausgaben:  Quotient ab DQTT                    *
;        :  Rest * 100  ab DDVD                 *
;Zerstört:  HL, DE, BC, AF, AF'                 *
;Subrout.:  CLR, RLDW, RRDW, DADD, DSUB,        *
;*********************************************

DDIV:   LD      HL,DDVD+1       ;++ Schiebe Dividend 1 Byte links
        LD      DE,DDVD         ; arithmetisch ( *100 ) +++
        LD      BC,DWL-1        ;(HL) --> (DE)=(HL)-1
                                ; HL+1, DE+1, BC-1
        LDIR                    ;Wiederhole, bis BC =0
        XOR     A               ;Lösche LB-Adresse des Dividenden
        LD      (DE),A          ;

        LD      HL,DQTT         ;++ Lösche Quotient ++
        CALL    CLR             ;

        LD      IX,INTVAR       ;Basisadresse für interne Variablen
        LD      (IX+0),1        ;DIVZähler :=1, da wenigstens eine
                                ;Division möglich ist !
        LD      (IX+1),1        ;ADDSUB :=1 (ungerade Parität
                                ;  --> Subtraktion )

SHIFT:  LD      HL,DDVS+DWL     ;++ Schiebe Divisor dezimal links
        CALL    RLDW            ;  arithmetisch ++

        INC     (IX+0)          ;DIVZähler +1
        LD      A,(HL)          ;Prüfe, ob MSD (Divisor) <> 0
        AND     0F0H            ;Falls ja, --> Linksbündig !
        JR      Z,SHIFT         ;Wiederhole, bis linksbündig

        LD      IY,DQTT-1       ;LB-Adresse (Quotient) - DWL
NDIV:   LD      A,(IX+1)        ;ADDSUB in Akku
        RRCA                    ;Rotiere LSB ins CF
        JR      NC,ADD          ;CF =0 --> ADD, sonst SUB

SUB:    LD      DE,DDVD+DWL     ;++ Subtraktion +++
        LD      BC,DDVD+DWL     ;LB- Adressen +1 der Operanden in
        LD      HL,DDVS+DWL     ;Doppelregister laden
        CALL    DSUB            ;( Rest(BC) - Divisor(HL) --> Rest(DE) )

        INC     (IY+DWL)        ;LSD (Quotient) +1
        JR      NC,SUB          ;Wiederhole, bis Ergebnis negativ

        DEC     (IY+DWL)        ;letzte Inkr. (Quotient) stonieren
        JR      OK1             ;überspringe die Addition !
```

```
ADD:    LD      A,(IY+DWL)      ;LSD (Quotient) :=9
        OR      09H             ; LSByte --> Akku, LSD:=9
        LD      (IY+DWL),A      ; Akku ---> LSByte (Quotient)

NADD:   LD      DE,DDVD+DWL     ;++ Addition +++
        LD      BC,DDVD+DWL     ;LB- Adressen +1 der Operanden in
        LD      HL,DDVS+DWL     ;Doppelregister laden
        CALL    DADD            ;( Rest(BC) + Divisor(HL) --> Rest(DE) )
        DEC     (IY+DWL)        ;LSD (Quotient) - 1
        JR      NC,NADD         ;Wiederhole, bis Ergebnis positiv

        INC     (IY+DWL)        ;letzte Dekr. (Quotient) stonieren

OK1:    INC     (IX+1)          ;ADDSUB +1 --> LSB Wechsel
        DEC     (IX+0)          ;DIVZähler - 1
        JR      NZ,NXT          ;Weiter, falls DIVZähler <> 0

        LD      DE,DDVD+DWL     ;++ Stoniereingaben +++
        LD      BC,DDVD+DWL     ;LB- Adressen +1 der Operanden in
        LD      HL,DDVS+DWL     ;Doppelregister laden

        LD      A,(IX+1)        ;ADDSUB in Akku
        RRCA                    ;Rotiere LSB ins CF

        RET     C               ;Fertig, falls letzte Operation
                                ; Addition war !

        CALL    DADD            ;sonst --> Stonieraddition ++
        RET

NXT:    LD      HL,DDVS         ;++ Schiebe Divisor dezimal
        CALL    RRDW            ;  logisch rechts ++

        LD      HL,DQTT+DWL     ;++ Schiebe Quotient dezimal
        CALL    RLDW            ;  arithmetisch links ++

        JR      NDIV            ;Verzweige zu nächster Division
```

```
;*********************************************************
;Aufgabe: "RRDW" rotiert ein Mehrbytewort (Dezimalzahl) *
;       : um 4 Bit ( 1 Nibble) logisch rechts           *
;Eingaben:  Dezimalzahl ab (HL)                         *
;Ausgaben:  Dezimalzahl / 100 ab (HL)                   *
;Zerstört:  HL, B, AF                                   *
;Subrout.:  ++ keine ++                                 *
;*********************************************************

RRDW:   LD      B,DWL           ;Bytezähler(B) := Dezimale Wortlänge
        XOR     A               ;Lösche Akku und Flags
RR:     RRD                     ;Rotiere (HL) dezimal rechts
        INC     HL              ;  ---> LSD in Low Nibble (A)
        DJNZ    RR              ;Wiederhole, bis alle Bytes rotiert

        RET
```

```
;*********************************************************
;Aufgabe: "RLDW" rotiert ein Mehrbytewort (Dezimalzahl) *
;       : um 4 Bit ( 1 Nibble) arithmetisch links       *
;Eingaben:  Dezimalzahl ab (HL)                         *
;Ausgaben:  Dezimalzahl / 100 ab (HL)                   *
;Zerstört:  HL, B, AF                                   *
;Subrout.:  ++ keine ++                                 *
;*********************************************************

RLDW:   LD      B,DWL           ;Bytezähler(B) := Dezimale Wortlänge
        XOR     A               ;Lösche Akku und Flags
RL:     DEC     HL              ;Rotiere (HL) dezimal links
        RLD                     ;  ---> MSD in Low Nibble (A)
        DJNZ    RL              ;Wiederhole, bis alle Bytes rotiert

        RET

;*********************************************************
;Aufgabe: "CLR" löscht einen Speicherbereich mit DWL    *
;       : Bytes ( z. Bsp. ein Mehrbytewort )            *
;Eingaben:  Bereichsanfang (Wortzeiger) im HL           *
;Ausgaben:  gelöschter Block ab (HL)                    *
;Zerstört:  HL, B, AF                                   *
;Subrout.:  ++ keine ++                                 *
;*********************************************************

CLR:    LD      B,DWL           ;Bytezähler(B) := Dezimale Wortlänge
CLR1:   LD      (HL),0          ;Lösche den durch HL adressierten
        INC     HL              ;Speicherplatz
        DJNZ    CLR1            ;Wiederhole, bis alle Bytes rotiert

        RET
```

```
*************************************************************
* +++ "SUBTR" : DERG = DOP1 - DOP2   +++
*Eingabe: Operand 1 ab DOP1
*       : Operand 2 ab DOP2
*Ausgabe: Ergebnis ab DERG
*************************************************************
*
SUBTR   SED             ;Dezimalmodus !
        LDX  #DWL-1     ;Dezimalzaehler := DWL -1
        SEC             ;CF:=1 fuer SBC
*
* Subtraktionsschleife, beginnen beim LB
SUBTR1  LDA  DOP1,X     ;++ byteweise Subtraktion ++
        SBC  DOP2,X     ;( DOP1 -DOP2 -moegl. Uebertrag
        STA  DERG,X     ;  --> DERG )
        DEX             ;naechst hoeheres Byte !
        BPL  SUBTR1     ;Wiederhole fuer alle Bytes
        RTS
```

```
*************************************************************
* +++ "ADDI" : DERG = DOP1 + DOP2   +++
*Eingabe: Operand 1 ab DOP1
*       : Operand 2 ab DOP2
*Ausgabe: Ergebnis ab DERG
*************************************************************
*
ADDI    SED             ;Dezimalmodus !
        LDX  #DWL-1     ;Dezimalzaehler := DWL -1
        CLC             ;CF:=0 fuer ADC
*
* Additionsschleife, beginnen beim LB
ADDI1   LDA  DOP1,X     ;++ byteweise Addition ++
        ADC  DOP2,X     ;( DOP1 +DOP2 +moegl. Uebertrag
        STA  DERG,X     ;  --> DERG )
        DEX             ;naechst hoeheres Byte !
        BPL  ADDI1      ;Wiederhole fuer alle Bytes
        RTS
```

```
*********************************************************
*    ++ "DMULT" : DPRD = DMTD * DMTR  ++
*Eingaben:  Multiplikand ab DMTD
*       :  Multiplikator ab DMTR
*Ausgaben:  Produkt ab DPRD
*********************************************************
*
DMULT   SED             ;In Dezimalmodus schalten
*
*Produkt loeschen
        LDX  #DWL-1     ;Wortlaenge in X-Reg.
        LDA  #$00       ;Akku loeschen
MULT1   STA  DPRD,X     ;(DPRD+X):=00
        DEX             ;Naechst hoeheren Wert
        BPL  MULT1      ;wiederhole bis fertig
*
*Multiplikatorziffer testen
        LDX  #$01       ;X-Zeiger auf hoechsten Wert
MULT0   LDA  DMTR,X     ;Byte (2 Ziffern) in Akku
        LSR             ;
        LSR             ;
        LSR             ;MSD des Multiplikatorbytes
        LSR             ;in unteren Nibble schieben
        BEQ  MULT2      ;springe wenn Null
        JSR  ADD        ;+++ Addiere (Nibble) mal +++
MULT2   JSR  SPL        ;+++ schiebe Produkt ein Nibble links +++
        LDA  DMTR,X     ;LSD des Multiplikatorbytes testen
        AND  #$0F       ;oberes Akkunibble ausblenden
        BEQ  MULT3      ;springe wenn Null
        JSR  ADD        ;Addiere (Nibble) mal
MULT3   INX             ;Wortzaehler (X) +1
        CPX  #DWL       ;Wort abgearbeitet?
        BEQ  ENDE       ;Springe wenn ja
        JSR  SPL        ;schiebe Produkt 1 Nibble links
        BNE  MULT0      ;wiederhole Multiplikatortest
ENDE    NOP             ;** Platz fuer Aufruf einer
        NOP             ; benutzereigenen Rundungsroutine
        NOP             ;
*
*schiebe Produkt ein Byte rechts
*   Kommataanpassung
        LDX  #DWL-1     ;Beginne beim niederwertigen Byte:
        LDY  #DWL-2     ;(DPRD+Y) ---> (DPRD+X)
ENDE1   LDA  DPRD,Y     ;Quelle (DPRD+Y) ueber Akku
        STA  DPRD,X     ; in Ziel (DPRD+X)
        DEX             ;naechstes hoeherwertiges Byte
        DEY             ; adressieren
        BPL  ENDE1      ;Wiederhole fuer alle Bytes
        LDA  #$00       ;Loesche MSByte des Produktes,
        STA  DPRD       ;da nur positive Zahlen !
        RTS             ;Fertig !
```

```
*
*Unterprogramm SPL:
*  +++ Schiebe Produkt dezimal arithmetisch links +++
SPL     LDY  #$04       ;1 Nibble (4) mal schieben
        STX  ZAEHL      ;Distanzzaehler in ZAEHL retten
SHIFT0  CLC             ;CF:=0
        LDX  #DWL-1     ;Beginne beim niederwertigen Byte:
SHIFT1  ROL  DPRD,X     ;  Rotiere Produkt 1 Bit links
        DEX             ;  mit 0 --> LSB
        BPL  SHIFT1     ;Wiederhole fuer gesamtes Produkt
        DEY             ;ZAEHLer - 1
        BNE  SHIFT0     ;Wiederhole, bis ZAEHLer = 0
        LDX  ZAEHL      ;Distanzzaehler wiederholen
        RTS
*
*Unterprogramm ADD
*   DPRD = DPRD +DMTD
ADD     TAY             ;Dezimalz. (Y) := Multiplikatorziffer
        STX  ZAEHL      ;Distanzzaehler(X) in ZAEHL retten
ADD1    LDX  #DWL-1     ;LB (Wort) adressieren
        CLC             ;loesche CF
ADD2    LDA  DPRD,X     ;((DPRD+X) + (DMTD+X) + CF)
        ADC  DMTD,X     ;  --> (DPRD+X)
        STA  DPRD,X     ;
        DEX             ;naechstes Byte
        BPL  ADD2       ;wiederhole fuer alle Bytes
        DEY             ;Dezimalzaehler - 1
        BNE  ADD1       ;wiederhole bis Y = 00
        LDX  ZAEHL      ;Distanzzaehler aus ZAEHL laden
        RTS
```

```
*********************************************************
*  ++ "DDIV" : DQTT = DDVD / DDVS  ++
*Eingabe:  Dividend ab DDVD
*       :  Divisor  ab DDVS
*Ausgabe:  Quotient ab DQTT
*       :  Rest * 100 ab DDVD
*********************************************************
```

```
*
DDIV    SED             ;Dezimalmodus
*
*multipliziere Dividend mit 100
        LDX  #$01       ;X-Zeiger auf hoechsten dezimalen Wert
DIV1    LDA  DDVD,X     ;Byte in Akku laden
        DEX             ;X-Zeiger auf naechst tiefere Adresse
        STA  DDVD,X     ;speichere hier
        INX             ;zeige auf naechst
        INX             ;  niederen dezimalen Wert
        CPX  #DWL       ;solange bis Wort abgearbeitet
        BNE  DIV1
        DEX             ;X-Zeiger auf die beiden Nachkommastellen
        LDA  #$00       ;lade 00 in den Akku um die beiden
        STA  DDVD,X     ;Nachkommastellen zu loeschen.
*
*loesche Quotient
DIV2    STA  DQTT,X     ;Akku immer noch Null, X auf Wortende
        DEX             ;zeige auf das naechste Byte
        BPL  DIV2       ;wiederhole bis Wortende
*
*Divisionszaehler u. ADDSUB initialisieren
        LDX  #$01       ;Initialwert in X-Reg
        STX  DIVZ       ;speichere X in Divisionszaehler
        STX  ADDSUB     ;und ADDSUB
*
*Divisor rechtsbuendig schieben
DVSL    LDY  #$04       ;Y=Zaehler fuer BCD-Schieben (4 mal)
DVSL0   CLC             ;CF:= 0, so dass LSD 00 wird
        LDX  #DWL-1     ;X-Zeiger auf Wortende
DVSL1   ROL  DDVS,X     ;schiebe (DDVS+X) links
        DEX             ;zeige auf naechstes Byte
        BPL  DVSL1      ;u. springe bis Wort abgearbeitet
        DEY             ;erniedrige "BCD-schiebezaehler"
        BNE  DVSL0      ;springe wenn noch nicht 4 mal geschoben
        INC  DIVZ       ;Divisionszaehler erhoehen
        LDA  DDVS       ;hoechstes Byte des Divisors in den Akku
        AND  #$F0       ;blende LSD aus
        BEQ  DVSL       ;wiederhole, falls MSD noch Null
*
*Division beginnt
DIV3    LDA  ADDSUB     ;lade ADDSUB in den Akku und teste
        AND  #$01       ;Test ob ADDSUB gerade oder ungerade
        BEQ  DADD       ;springe in Addition wenn gerade
*
*Subtraktionsteil
DSUB0   SEC             ;CF:= 1, kein Borrow fuer Subtraktion
        LDX  #DWL-1     ;Beginne beim niederwertigen Byte
DSUB1   LDA  DDVD,X     ;<(DDVD+X) - (DDVS+X) - CF>
        SBC  DDVS,X     ;  --> (DDVD+X)
        STA  DDVD,X     ;
        DEX             ;zeige auf naechstes Byte
        BPL  DSUB1      ;springe wenn X >= 00
        INC  DQTT+DWL-1 ;incrementiere LSD des Quotienten
        BCS  DSUB0      ;wiederhole, bis Ergebnis negativ
*Zaehler auf Quotienten aufaddieren
        DEC  DQTT+DWL-1 ;letzte Inkr. (Quotient) stornieren
        BCC  DIV4       ;springt immer!
*
*Additionsteil
DADD    LDA  #$09       ;9 in LSD (Quotient)
        ORA  DQTT+DWL-1 ;wobei MSD unveraendert bleibt
        STA  DQTT+DWL-1 ;
DADD0   CLC             ;CF = 0 fuer Addition
        LDX  #DWL-1     ;beginne beim niederwertigen Byte
DADD1   LDA  DDVD,X     ;<(DDVC+X) + (DDVS+X) +CF>
        ADC  DDVS,X     ;  --> (DQTT+X)
        STA  DDVD,X
        DEX             ;zeige auf naechstes Byte
        BPL  DADD1      ;springe wenn noch nicht fertig
        DEC  DQTT+DWL-1 ;LSD (Quotient) - 1
        BCC  DADD0      ;wiederhole bis Ergebnis pos.
        INC  DQTT+DWL-1 ;letzte Dekr. (Quotient) stornieren
*
*neue Addition bzw. Subtraktion vorbereiten (od. ENDE)
DIV4    INC  ADDSUB     ;ADDSUB +1 (Vorzeichenwechsel)
        DEC  DIVZ       ;DIVzaehler - 1
        BEQ  ENDE       ;weiter wenn DIVzaehler <> 0
*
*Divisor logisch rechts schieben
        LDY  #$04       ;dezimales rechtsschieben
DVSR0   CLC             ;das MSD muss 00 werden
        PHP             ;CF retten
        LDX  #$00       ;beginne beim MSD des Wortes
DVSR1   PLP             ;hole CF
        ROR  DDVS,X     ;rotiere (DDVS+X) rechts CF->MSB, LSB->CF
        PHP             ;rette CF
        INX             ;zeige auf naechstes Byte
        CPX  #DWL       ;DWL Byte abgearbeitet?
        BNE  DVSR1      ;wiederhole falls nein
        PLP             ;raeume Stack auf
        DEY             ;schon 4 mal geschoben?
        BNE  DVSR0      ;wiederhole falls nein
*
*Quotient dezimal links schieben
        LDY  #$04       ;dezimales linksschieben
QSHL0   CLC             ;CF:=0, so dass LSD 0 wird
        LDX  #DWL-1     ;beginne mit hoechstem Byte
QSHL1   ROL  DQTT,X     ;CF->LSB(Quotient),LSB(Quotient)->CF
        DEX             ;zeige auf naechstes Byte
        BPL  QSHL1      ;springe bis X < 00
        DEY             ;schon 4 mal geschoben?
        BNE  QSHL0      ;wiederhole wenn noch nicht 4 mal
        JMP  DIV3       ;naechste Addition/Subtraktion
*
*Division fertig
ENDE    LDA  ADDSUB     ;teste, ob beim letzten mal
        AND  #$01       ;addiert oder subtrahiert wurde
        BNE  ENDE1      ;springe wenn addiert wurde
*
*Storniere letzte Subtraktion
        CLC
        LDX  #DWL-1     ;zeige auf niedrigstes Byte
ENDE0   LDA  DDVD,X     ;<(DDVD+X) + (DDVS+X) + CF>
        ADC  DDVS,X     ;  --> (DDVD+X)
        STA  DDVD,X     ;
        DEX             ;zeige auf naechstes Byte
        BPL  ENDE0      ;springe wenn X >= 0
*
ENDE1   RTS             ;Fertig !
```


Auch als Bausatz

PROTON *intelligente* Tastaturen

High Quality - Low Cost

PROTON-Tastaturen sind mit Keyswitches von **Futuba,** weltführender Hersteller, aufgebaut. Diese Keyswitches werden auch von führenden Terminal-Herstellern wie **Lear-Siegler** und **Televideo** eingesetzt.

PROTON-Tastaturen werden in Holland hergestellt. Neben den standardmäßigen Tastaturen sind auch kundenspezifische Tastaturen preislich sehr attraktiv, auch in kleineren Stückzahlen.

Befestigungsplatte aus Stahlblech. Auf die Platine wird keine mechanische Kraft ausgeübt: Ein zuverlässiges Funktionieren ist damit gewährleistet.

ASCII-Encoder mit wählbarer Tastenbelegung und **parallelem** und **seriellem** ASCII-Ausgang mit wählbaren Schittstellen-Daten. Größte Flexibilität: Anpassung an jeden Computer möglich.

16 programmierbare Funktiontasten. Unter jede der Funktiontasten können Sie einen String bis zu 15 Charactern in EPROM ablegen, so daß Änderungen - auch nachträglich - problemlos sind. Von der Tastatur aus können diese Strings vorübergehend überschrieben werden (in das interne RAM). Standardmäßig sind die am häuftigst benutzten BASIC-Befehle abgelegt.

		Incl. MwSt.	excl. MwSt.
KB-2 Matrixtastatur 10 x 10	DM	239,40	210,00
KB2E-G komplett gebaute Tastatur mit Encoder und Gehäuse.	DM	416,10	365,00
KB2E-B wie KB2E-G, jedoch als Bausatz	DM	324,90	285,00
QWERTZ Umbausatz auf deutsche Tatastenbelegung incl. EPROM.	DM	27,82	24,40
APPLECABLE Flachband-kabel (1 m) mit zwei 16-pol. IC-Steckern für Apple-Anschluß	DM	38,76	34,00

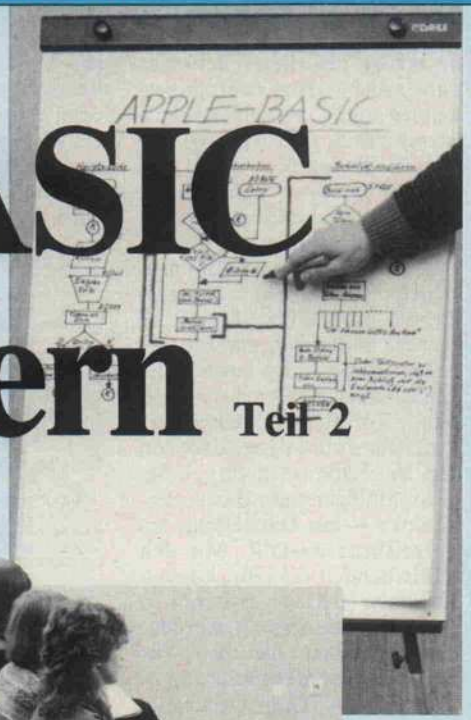
Bitte fordern Sie sofort das Informationsmaterial an!

OEM's fragen Sie gezielt an!

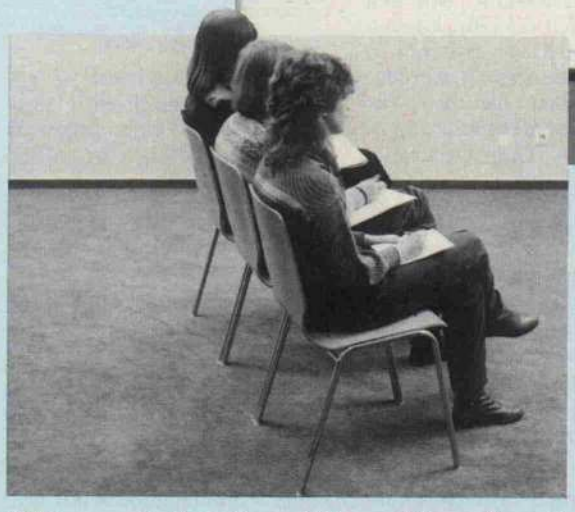
PROTON

In den Preisen sind 14% MwSt. enthalten. Soweit nichts anderes vereinbart, erfolgt der Versand gegen Nachnahme. Pauschalbetrag für Versand und Verpackung 9,50 DM.

TEEPE GmbH
Vorm Tor 8 / D-6395 Weilrod
Telefon 06083/2329/553



BASIC intern Teil 2



Was nicht im Handbuch steht

Heinz-Peter Heidinger

Der erste Teil dieses Beitrages gab einen Überblick über die von BASIC benutzten Speicherabschnitte. Das Augenmerk lag auf der speicherinternen Struktur des Programmtextes im Microsoft-BASIC. Im zweiten Teil sollen nun die Speicherabschnitte näher betrachtet werden, die der Interpreter beansprucht.

Bei genauer Betrachtung der Arbeitsweise von BASIC, läßt sich das, was gemeinhin als Interpreter angesehen wird, in mehrere Teile untergliedern. Wenn Sie nämlich die Programmzeilen eingeben, ist zunächst gar keine Interpretation im Spiel. (Interpreter, die schon bei der Eingabe die Syntax prüfen, werden hier ausgeklammert.) Die Routinen, die das Eingeben und Ändern von Programmzeilen gestatten, bilden den Editor.

Im ersten Teil dieses Artikels wurde dargestellt, daß ein BASIC-Programm im Speicher eine andere Form hat, als Sie es beim Listen auf dem Schirm sehen können. Es wird für den Interpreter in dieses spezielle Format übersetzt (tokisiert) und zur Einsichtnahme für den Benutzer (LIST) wieder rückübersetzt (detokisiert). Folglich enthält der BASIC-Interpreter auch einen Übersetzer/Rückübersetzer, der allerdings eng mit dem Editor verflochten ist.

Diesen Erkenntnissen folgend gliedert sich BASIC also wie in Bild 2 dargestellt. Dabei greift der obere Funktionsblock nur auf das Programmtextsegment zu, während alle anderen Speicherbereiche nur durch den eigentlichen Interpreter aufgebaut und manipuliert werden. Diesem 'Rest der (BASIC-) Welt' soll heute besondere Aufmerksamkeit gelten.

Ein leichter Einstieg in die Thematik ergibt sich, wenn man zunächst einen Blick auf die Typcode-Tabelle (Tabelle 2 in c't 4/84, S. 46) wirft. Alle Variablen mit dem Anfangsbuchstaben 'A' tragen, dieser Tabelle entsprechend, den Typcode '02'. Da in unserem SETUP-Programm gleichfalls alle Variablen mit diesem Anfangsbuchstaben als ganzzahlig deklariert sind (130 DEFINT A), fällt der gedankliche Schluß nicht schwer: der Typcode '02' kennzeichnet eine INTEGER-Variable. Mit dieser Erkenntnis gerüstet, ist auch der Rest der Tabelle schnell entschlüsselt. Alle 'C'-Variablen sind als 'einfach genau' definiert, und ihnen entspricht der Typcode '04'. Für 'doppelt genaue' Variablen (DEFDBL E) gilt, wie dem fünften Byte der Tabelle zu entnehmen ist, der Typcode '08' und für Strings (DEFSTR G) der Code '03'. Die übrigen Stellen der Tabelle enthalten für alle anderen Anfangsbuchstaben '04', das heißt, für alle Variablen, deren Daten-Typ nicht ausdrücklich definiert wurde, wird einfache Genauigkeit angenommen, sofern dem Variablen-Namen beim Aufruf keine Typenkennzeichnung (%, !, #, $) folgt. Diese Vereinbarung werden Sie aber sicher schon Ihrem BASIC-Handbuch entnommen haben.

Funktionsblock	Zugriff auf Speichersegment
Editor Übersetzer (Tokenizer)/ Rückübersetzer (Detokenizer)	Programmtext
Interpreter	Programmtext (nur lesend!) Variablentabelle ARRAY Variablentabelle Stack Stringspace

Bild 2. Gliederung des Microsoft-BASIC-Interpreters

Typcode und Variable

Es gilt also:

Typcode	Variablentyp
02	Integer
03	String
04	einfach genau
08	doppelt genau

Typcode?! Schön! ... und gut?

Es ist sicherlich leicht einsehbar, daß für den Wert einer Variablen mit doppelter Genauigkeit mehr Speicherplatz benötigt wird, als für den Wert einer Variablen mit einfacher Genauigkeit. Kurz gesagt: Jeder Wert einer Variablen belegt in der Variablen-Tabelle so viele Bytes, wie der Typcode angibt. Weil diese Tabelle außer dem Wert grundsätzlich auch den Typcode und den Namen der Variablen mitspeichert (2 Zeichen in MBASIC 4.x und dessen Abkömmlingen), beansprucht jede Variable folglich Typcode + 3 Bytes. Das bedeutet für den Interpreter, daß er den Wert einer Variablen immer ab Byte # 4 im entsprechenden Feld der Variablentabelle findet. In Versionen von MBASIC 5.x sind Variablennamen mit bis zu 40 Zeichen zugelassen. Hier werden alle Zeichen des Variablennamen festgehalten und zusätzlich die Information über die Länge des Namens, damit der Interpreter auch hier weiß, von welchem Byte an er den zugehörigen Wert findet (Tabelle 7).

Die Variablen-Tabelle

In Tabelle 6 ist die vollständige Variablentabelle zu sehen, die das SETUP-Programm aus dem ersten Teil dieses Beitrags angelegt hat. Wie beschrieben, verweist das erste Byte eines Eintrages immer auf den Daten-Typ der verwendeten Variable. Der Speicherauszug zeigt also zunächst mit '04', daß die vorliegende Variable einfache Genauigkeit hat und ihr korrespondierender Wert vier Byte Speicherplatz beansprucht. In den Bytes # 2 und # 3 wird in ASCII der Name der Variablen festgehalten (MBASIC 4.x), und zwar immer das zweite signifikante Zeichen zuerst. Falls der Variablenname nur aus einem Zeichen besteht, ist Byte # 2 immer '00'. Für das Beispiel heißt das, daß die vorliegende Variable den Namen mit dem Zeichencode 'C9h' trägt.

Allerdings ist 'C9h' ja kein

```
# DUMP  6DBC  6E36
6DB0                                  |04 00 C9 C7  *           ....*
6DC0  6A 00 00|02 00 41 00 7D|04 00 42 00 00 00 00|02  *j....A.ü..B.....*
6DD0  00 42 20 4E|04 00 43 E6 6E 32 90|04 00 44 1A D5  *.B N..C.n2...D..*
6DE0  5D 90|08 00 45 76 5B F3 A3 A2 79 6B 9B|08 00 46  *ü...EvA...yk...F*
6DF0  2B CA C0 23 8D B3 5F 9C|03 00 47 15 9F 6B|03 00  *+..#.._...G..k..*
6E00  48 1B BB 6B|02 49 41 39 30|04 49 43 66 E6 40 BE  *H..k.IA90.ICf.§.*
6E10  08 49 45 76 5B F3 A3 A2 79 6B 9B|03 49 47 04 FB  *.IEvA...yk..IG..*
6E20  BF|04 00 5B 00 00 00 84|04 00 59 00 00 00 84|04  *...X.....Y.....*
6E30  00 5A 00 00 00 84                               *.Z....          *
```

Tabelle 6. Variablen-Tabelle von SETUP/BAS

ASCII-Zeichen, weil der Code größer als 7Fh ist! Und dennoch verbirgt sich dahinter der ASCII-Code eines Variablennamen, und zwar der Name 'I' (49h). Offenbar zeigt sich gleich bei erster näherer Betrachtung eine Besonderheit, denn es handelt sich bei diesem Zeichencode um ein sogenanntes 'marked ASCII', bei dem das Bit#7 einfach auf '1' gesetzt ist, um damit etwas 'Besonderes' zu kennzeichnen. Mit Blick auf den BASIC-Text von SETUP findet man auch gar keine Variable 'I'. Dieser Name tritt nur bei einer benutzerdefinierten Funktion (DEF FN I(A,B) = ...) auf. Dadurch ist die (richtige) Schlußfolgerung wieder nahegelegt: Variablennamen mit Bit#7 = '1' im ersten Zeichen des Namens verweisen auf eine 'Benutzerfunktion' (user function) und verlangen somit besonderes Verhalten des Interpreters. Die beiden folgenden Bytes stellen dann keinen Wert dar, sondern geben die Adresse im Programmtext (LSB/MSB-Format) an, an der der Interpreter die eigentliche Funktion findet, die beim Aufruf von FN I(x,y) auszuführen ist.

Dem Typcode entsprechend liefert die Beispiel-Funktion 'I' immer ein einfach genaues Ergebnis, und das Wert-Feld beansprucht vier Bytes. Da jedoch die Referenz-Adresse nur zwei Byte beansprucht, muß der Rest des Wert-Feldes — der Typcode-Vereinbarung folgend — hinter der Adresse mit '00'-Bytes aufgefüllt werden. Eine Wertübergabe findet nicht über das Wert-Feld statt, weil man einer Funktion ja prinzipiell keinen Wert zuweisen kann und dies auch die Programmtextadresse überschreiben würde. Sie liefert immer nur ein Ergebnis, das nach der Kalkulation in einem Arbeitspuffer des BASIC-Interpreters vorliegt und von dort übernommen wird.

TC	V1 V2	LN	V3 ... VN	TC-Bytes-Wertfeld
a.	b.	c.	d.	

a.) Typcode der Variable

b.) Die beiden ersten Zeichen des Namens in der richtigen Reihenfolge (also nicht V2 V1 wie in MBASIC 4.x).

c.) Längeninformation über den Variablennamen; (Anzahl der LN noch folgenden Zeichen des Namen)

d.) weitere Zeichen des Variablennamen. Aber als 'marked ASCII', das heißt Bit#7=1. Diese werden erst zu Unterscheidung herangezogen, wenn mit V1 und V2 kein eindeutiger Variablenbezug erkannt wird.

Tabelle 7. Variablen-Feld im MBASIC 5.x

Abschließend kann man feststellen, daß man Verweise auf Benutzerfunktionen in der Variablen-Tabelle findet, obwohl diese selbst eigentlich gar keinen Datentyp darstellen. Vielmehr erkennt der Interpreter an der besonderen Kennzeichnung des Variablennamens, daß er eine Art 'impliziten GOSUB' auszuführen hat, wenn er bei der Programmausführung auf einen Funktionsnamen stößt.

Numerische Variable

Microsoft-BASIC stellt für die Verarbeitung numerischer Daten grundsätzlich zwei verschiedene Datentypen zur Verfügung: Ganzzahlen (INTEGERS) und Fließkommandozahlen (reelle Zahlen, REALS). Letztere sind mit 'einfacher Genauigkeit' (single precision, SNG) ≙ 6 gültigen Stellen oder 'doppelter Genauigkeit' (double precesion, DBL) ≙ 16 gültigen Stellen darstellbar.

INTEGERS

In Form von Zeilennummern sind ganzzahlige Werte schon im ersten Teil dieses Beitrages aufgetaucht. Hier war zu erkennen, daß Ganzzahlen intern immer durch zwei Byte (16 Bit) dargestellt werden, und dies wird inzwischen auch durch den Typcode bestätigt. Anders als die Zeilennummern, die nur als vorzeichenlose Ordnungszahlen ('natürliche Zahlen') dienen, muß für arithmetische Zwecke an Integers ein weiterer Anspruch gerichtet werden: Es müssen auch Zahlen mit negativem Vorzeichen darstellbar sein. Das führt dazu, daß von den 16 verfügbaren Bits, eines — sinnfälligerweise das äußerste linke — zur Darstellung des Vorzeichens ausfällt. Mit den verbleibenden 15 Bit können nur noch Beträge zwischen 0 und 32767 dargestellt werden. Zunächst scheint dies die — für den Anfänger meist seltsam anmutende — Grenzwertfestlegung zu erklären. Als größter Betrag für negative Zahlen, ist dem Handbuch aber '32768' zu entnehmen. So ist auch 8000h = 1000/0000/0000/0000b nicht gleich dem wenig sinnvollen Ausdruck —0, sondern entspricht —32767. Für die Betragsermittlung negativer Ganzzahlen gelten offenbar andere Gesetzmäßigkeiten (siehe Bild 3).

Intern werden vorzeichenbehaftete Integerzahlen immer im Zweierkomplement dargestellt, wenn das Vorzeichenbit gesetzt ist. Man merkt sich erst, daß die Zahl ein negatives Vorzeichen trägt und bildet dann im ersten Zwischenschritt das Einerkomplement. Das heißt, jedes Bit der vorliegenden Zahl wird invertiert:

Beispiel: F842h

1111/1000/0100/0010b
EK = 0000/0111/1011/1101b

Das Zweierkomplement wird nun (zweiter Schritt) dadurch gebildet, das zum 'EK' eins addiert wird; das Ergebnis ist der Betrag der negativen Zahl:

```
   EK = 0000/0111/1011/1101b
+   1 = 0000/0000/0000/0001b
= ZK = 0000/0111/1011/1110b
       0    7    B    E    h = 1998d
```

Da der Ausgangswert — gekennzeichnet durch ein gesetztes Bit#15 — ein negatives Vorzeichen trug, ist also: F842h = —1998d.

Nachdem die Frage des Wertungsspektrums von Integer-Zahlen nun weitgehend geklärt ist, soll hierzu abschließend noch deren speicherinternes Format in der Variablentabelle sondiert werden. Die im SETUP-Programm verwendeten INTEGERS 'A' (implizit), 'B' (explizit, durch Typenkennzeichner) und 'AI' findet man in der Variablentabelle auf den Speicherplätzen: 6DC3h, 6DCFh und 6E04h, wie am Typcode '02' leicht erkennbar

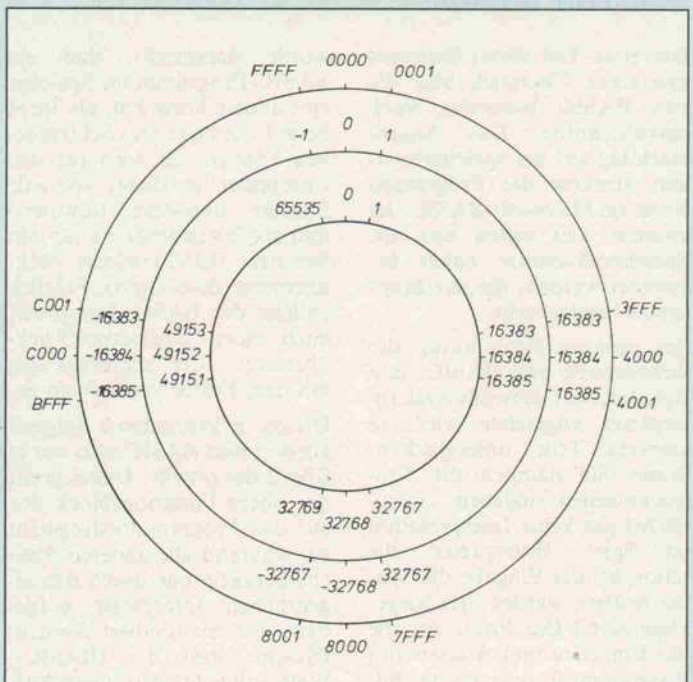


Bild 3. 'Zahlenzirkel'

ist. Für die ersten drei Bytes jedes Feldes gilt das bereits Gesagte. Im Wertfeld (Bytes 4—5) steht dann der binäre Wert, welcher der entsprechenden Variable zugewiesen wurde — und zwar immer das niederwertige Byte zuerst. Somit gilt:

Byte#4 + 256*Byte#5,

wenn Byte#5 < 128 (80h) ist und damit positives Vorzeichen signalisiert, oder:

(Byte#4—256) + 256*
(Byte#5—255)
entspr. (Byte#4 + 256 *
Byte#5) —65536,

wenn Byte#4 > 127 ist und durch ein gesetztes MSB die Zweierkomplementbildung fordert.

REALS

Reals sind Zahlen, die auch Nachkomma-Anteile mitführen, und die im allgemeinen auch als Gleit- oder Fließkommandozahlen bezeichnet werden. Sie können in MBASIC mit einfacher Genauigkeit (Typcode '04') oder doppelter Genauigkeit (Typcode '08') gehandhabt werden. Hier stellt sich also auch die Frage der darstellbaren Genauigkeit. Wie bei den Integers sollen aber zunächst die Fragen des internen Darstellungsformates und der Darstellungsgrenzen geklärt werden. Geschehen soll dies am Beispiel einfach genauer Variablen, weil sich die prinzipiellen Überlegungen leicht auf doppelte Genauigkeit übertragen lassen.

Grundsätzlich lassen sich alle Zahlen im Exponentialformat darstellen. Dabei gibt es eine Mantisse, die den numerischen Wert der Zahl repräsentiert und einen Exponenten, der die Größenordnung der Zahl angibt.

Für das Dezimalsystem gilt allgemein, daß die Mantisse immer durch eine Zahl dargestellt wird, die größer oder gleich 1, aber kleiner als 10 ist. Alle anderen Stellen der Zahl sind Nachkommastellen. Die Zahl 987,654 bietet sich demnach mit der Mantisse 9,87654 dar. Es muß nun noch ein Wert angegeben werden, mit dem jede Stelle der Mantisse zu multiplizieren ist. Dieser Wert ist der Exponent, der hier 10^2 (=100) lauten müßte:

Mantisse/Exponent
$9{,}87654 * 10^2$

9,0	* 100	=	900,0	+
0,8	* 100	=	80,0	+
0,07	* 100	=	7,0	+
0,006	* 100	=	0,6	+
0,0005	* 100	=	0,05	+
0,00004	* 100	=	0,004	+
		=	987,654	

Aus der Variablentabelle läßt sich zunächst die allgemeine Form einer Gleitkommazahl im Speicher ersehen:

04 V2 V1 N1 N2 N3 N4 (SNG)
08 V2 V1 N1 N2 .. N8 (DBL).

Hierbei enthält das letzte Byte (N4 bzw. N8) den Exponenten. Da die Mantisse Byte-weise codiert ist und mit jedem Byte 256 (2^8) Werte dargestellt werden können, ist der jeweilige Stellenwert eine Potenz von 256. Diese grundsätzlichen Zusammenhänge sind in Bild 4 wiedergegeben und gelten für doppelt genaue Variable — bei entsprechender Erweiterung dieser Berechnungsvorschrift für acht Byte — gleichermaßen.

Natürlich müssen auch Reals mit positivem und negativem Vorzeichen dargestellt werden können. Analog zu den Integers übernimmt wieder das MSB des höchstwertigen Bytes der Mantisse (N3 bzw. N7) die Funktion, das Vorzeichen zu markieren. Das heißt, wenn es gesetzt ist, gilt (N3—128) und das Ergebnis erhält ein negatives Vorzeichen. Mit den bisherigen Erkenntnissen lassen sich nun auch die kleinste und die größte Zahl berechnen:

Positiv:

N1 N2 N3 N4 = FF FF 7F FF = 255 255 127 255

$2^{126} * (\ 255 \quad /128$
$+255 \quad /128*256$
$+127 \quad /128*256^2)$
$= 1{,}7141 * 10^{38}$

Negativ:

N1 N2 N3 N4 = FF FF FF FF = 255 255 255 255

$2^{126} * (\ 255 \quad /128$
$+255 \quad /128*256$
$+255—128/128*256^2)$
$= -1{,}7141 * 10^{38}$

Ungeklärt ist bis jetzt noch immer der in Bild 4 angegebene Versatzwert 'q'.

Bezugnehmend auf die vorangegangene Berechnung der Darstellungsgrenzen ist aber leicht erkennbar, daß dieser den Wert —129 haben muß. Da auch Bruchzahlen mit Beträgen kleiner als 1 dargestellt werden sollen, ist es folglich auch notwendig, den Exponenten mit negativem Vorzeichen abbilden zu können.

Statt der sonst üblichen Zweierkomplement-Bildung wendet man aber ein anderes Verfahren an, um das Vorzeichen des Exponenten zu erhalten. Man rechnet zum Exponenten-Byte N4 beziehungsweise N8 konsequent 81h (129, vorzeichenlos) hinzu. Wenn N4/N8 vorher schon größer als 7Fh war, wird sich dabei immer ein Wert ergeben, der zwischen 01—80h liegt. Beim späteren Rückrechnen der Zahl wird dieser Versatzwert vom Exponenten-Byte wieder abgezogen, und ein prozessorinternes Statusbit (CARRY) zeigt nach dieser Operation an, ob diese Subtraktion ohne 'borgen' möglich war oder nicht. Dadurch ist ein unmittelbarer Rückschluß auf das Vorzeichen möglich.

Dem besonders kritischen Leser wird sicher schon ein Mangel der Berechnungsvorschrift aufgefallen sein, der in Bild 4a besonders deutlich hervortritt. Es ist mit diesem Algorithmus nicht möglich, den Wert '0' mathematisch exakt abzubilden, selbst wenn alle Bytes der speicherinternen Darstellung '00' sind. Zwar werden dann alle Summenglieder, die ein Mantissen-Byte enthalten gleich 0, das letzte Glied der Summe stellt sich aber als $2^{(0-129)} = 2^{-129} = 1{,}45 * 10^{-39}$ dar. Dies ist zwar ein sehr kleiner Wert — aber nicht gleich Null! BASIC geht in diesem Fall so vor, daß der Bytefolge '00 00 00 00' der Wert 0 'aufgezwungen' wird. Als kleinster Betrag, der speicherintern darstellbar ist, ergibt sich somit 2^{-128} ($2{,}93875 * 10^{-39}$) mit der Bytefolge '00 00 00 01'. Allerdings weigert sich der BASIC-Interpreter beharrlich, eine direkte Zuweisung 'A=2↑—128' in dieser Folge zu codieren. Statt dessen findet man '00 00 00 00', und eine Abfrage 'PRINT A' wird folglich auch mit '0' quittiert. Ein annähernd richtiges Ergebnis ereicht man mit einer List: 'A=2↑—127 * 2↑—1'. Dieser Ausdruck ist mathematisch äquivalent zu 2^{-128}, zeigt aber im Speicher noch immer nicht die korrekte Bytefolge, sondern '0C 00 00 01' — also einen Rundungsfehler von $1{,}43 * 10^{-6}$. Ein absolut korrektes Ergebnis erhält man aber, wenn folgendermaßen formuliert wird:

'A=1: FOR I=1 TO 128:
A=A * 2↑—1: NEXT'

Dies bedeutet zwar ebenfalls 2^{-128}, liefert aber die geforderte Bytekette '00 00 00 01'.

a) $$\frac{2^{N4-q} \times N3}{2^7 \times (2^8)^0} + \frac{2^{N4-q} \times N2}{2^7 \times (2^8)^1} + \frac{2^{N4-q} \times N1}{2^7 \times (2^8)^2} + 2^{N4-q}$$

faktorisiert:

b) $$2^{N4-q} \times \left(\frac{N3}{2^7 \times (2^8)^0} + \frac{N2}{2^7 \times (2^8)^1} + \frac{N1}{2^7 \times (2^8)^2} + 1 \right)$$

Bild 4. Berechnungsvorschrift für 'Reals'

Stellengenauigkeit

Schaut man wieder auf die Berechnungsvorschrift in Bild 4, so wird erkennbar, daß der kleinste Bruchanteil dieser Summenformel durch das Summenglied mit dem größten Teiler gebildet wird (niederwertigstes Mantissen-Byte N1). Wird nun N1 gleich '1' gesetzt und der Teiler dezimal umgerechnet, ist am Teiler-Exponenten unmittelbar die erreichbare Stellengenauigkeit ablesbar. Die jeweiligen Werte, die die Ausdrücke für N1=1 liefern, sind die kleinstmögliche Einheit, mit der der Variablenwert wachsen könnte. Sie ergeben sich:

für SNG:
$1/(2^7 * (2^8)^2)$
$= 1/(8{,}3 * 10^6$ (6 Stellen!)
kleinste Einheit: $1{,}2 * 10^{-7}$

für DBL:
$1/(2^7 * (2^8)^6)$
$= 1/(3{,}6 * 10^{16})$ (16 Stellen!)
kleinste Einheit: $2{,}78 * 10^{-17}$

Soweit die Untersuchungen über die Speicherung und Codierung numerischer (unstrukturierter) Daten.

Im dritten Teil der Serie wird die Aufmerksamkeit auf Strings und den strukturierten Datentyp 'ARRAY' gerichtet sein.

(Fortsetzung folgt)

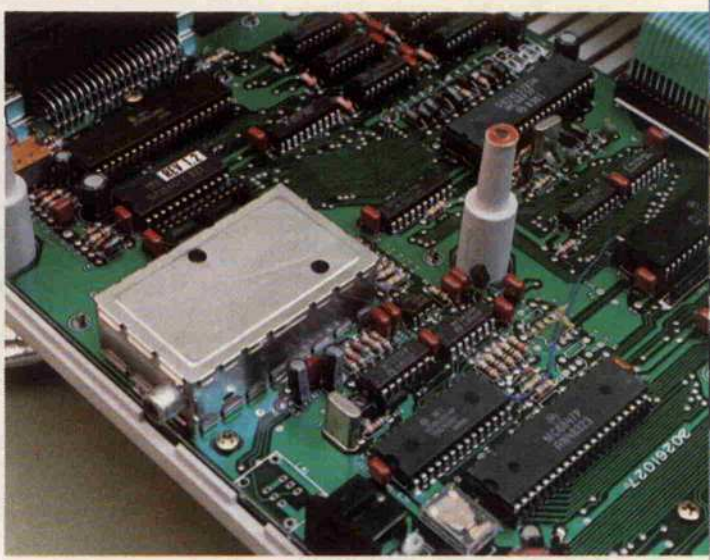
TRS-80 Colour Computer 2

Kein großer Wurf

Eckart Steffens

Preise

Colour Computer 16 K	DM	748,—
dito, mit extended Basic	DM	948,—
dito, 64 K mit extended Basic	DM	1 250,—



Art, Aufmachung und Preis weisen den Nachfolger von Tandys erstem Colour Computer eindeutig als einen Pendanten zu den bereits auf dem Markt befindlichen Modellen anderer Anbieter aus. Wie wird sich der Colour Computer 2 gegen die Vielfalt der Konkurrenz behaupten können? Bei näherem Hinsehen kamen uns einige Zweifel ...

Die Spannung beim Auspacken eines neuen Computers ist jedesmal wieder hoch, gilt doch auch hier der Satz vom ersten Eindruck. Der TRS-80 zieht sich gut aus der Affäre: sauber verpackt, stabiles, hellbeiges genopptes Gehäuse, angenehmes Styling und — sehr erfreulich — eine schöne Tastatur, die sofort als übersichtlich eingestuft wird und auch ein positives Schreibgefühl vermittelt. Die dicken Anleitungsbücher (deutsch!) legen wir erstmal zur Seite und wagen den Blick auf die Rückseite des Gerätes. Da finden wir den Netzschalter und die Netzzuführung, Anschlüsse für Joysticks und den seriellen Bus (Drucker, Floppy) sowie Kassettenrecorder. Alles DIN-Buchsen. Ein Cinch-Ausgang vom Modulator und eine leichtgängige Reset-Taste.

Computer à la carte ...

Wenn das Gerät ausgepackt ist, beginnt auch das Nachschauen, welche Konfiguration man denn nun hat: 4, 16 oder 64 KByte RAM, normales Basic oder Extended Basic. Wir hatten 16 K (14,6 KBytes free) und Normal-Basic. Nun, für 748 DM akzeptabel. Welche Befehle außer dem wohl auf fast allen Rechnern implementierten Standardsatz hat man dann zur Verfügung? Zum Beispiel diese: AUDIO, schaltet den Kassettenrecorderausgang auf den Fernseher; MOTOR, schaltet den Recordermotor ein oder aus; SOUND, mit Angabe der Tonhöhe und -dauer, erzeugt einen Ton; PRINT@, Bildschirmausgabe ab einer bestimmten Position; SET x,y,c, setzt Bildschirmpunkt x/y auf Farbe c; RESET x,y, löscht Punkt x/y; Point x/y, liefert eine Information, ob und in welcher Farbe ein Bildschirmpunkt gesetzt ist; JOYSTK, fragt die Stellung der Joysticks ab.

Die Bezeichnung Colour Computer deutet darauf hin, daß die Farbe eine wichtige Domäne dieses TRS-80 sei; der Befehlssatz unterstützt dies jedoch nicht sehr. Will man die fraglos vorhandenen farblichen (8 Farben + Schwarz) und grafischen (64 x 32 bis zu 256 x 192 Punkte) Möglichkeiten des Colour Computer 2 nutzen, sollte man unbedingt die Extended-Colour-Basic-Version bevorzugen. Erst dann läßt sich der Grafikmodus einfach per Befehl bestimmen, kann mit mit DRAW, LINE, CIRCLE, PAINT und SCREEN arbeiten und hat insgesamt 44 Befehle mehr zur Verfügung. Darunter auch PRINT USING und Tools wie DELETE und RENUM.

Nun wird's bunt

Ein prachtvolles Farbbild bietet sich dem Betrachter nach Inbetriebnahme der Maschine wahrlich nicht: Kleine, dünne, dunkle Buchstaben auf hellgrünem Grund, umrahmt von einer schwarzen Maske, präsentiert sich der Colour Computer. Da kein Monitorausgang vorhanden, ist man auf den gewöhnlichen Farbportable angewiesen.

Nach einer Viertelstunde flimmern die Augen. Schwarzweiß ist's um etliches besser, aber wozu braucht man dann einen Colour Computer? Daß die Farbe Hauptdomäne sein soll, demonstriert der Cursor: er blinkt nicht an und aus, sondern wechselt wie ein Chamäleon laufend seine Farbe. Sei es.

Also denn greifen wir mal in die Tasten und bringen die Maschine zum Arbeiten. Der Tandy schreibt alles im 'Caps Lock' Modus, das heißt auf dem Schirm erscheinen Großbuchstaben. Beim Drücken von 'Shift' passiert nichts — der Colour Computer hat keine Festgrafik.

Wo ist die Shift-Lock-Taste? Siehe da. Es gibt also Kleinbuchstaben — laut Handbuch durch Umschalten mittels 'Shift 0' zu erreichen. Gut, das haben wir. Doch während nun auf dem Drucker Kleinbuchstaben erscheinen, gibt's auf dem Schirm nur revers dargestellte Großbuchstaben.

Passen Sie auf mit diesem reversen Zeug, die Maschine versteht's nicht als Befehl. 'SN ERROR' (der vornehme Computer spricht in Abkürzungen, spart Speicherplatz!?) ist das höchste, was Sie daraufhin zu sehen kriegen. Also hoch mit dem Cursor in die vorherige Zeile und die Sache korrigiert. Das ist ja wohl einfach mit den schönen, großen und übersichtlichen Steuertasten. Denkste. Der Cursor ruckt und rührt sich nicht; zwei der Pfeiltasten drucken eine eckige Klammer auf den Schirm und die dritte löscht die letzte Eingabe, mit Shift sogar gleich mehrere Zeilen weg. Cursor, Cursor ... warten Sie mal, das haben wir gleich ... Nein, im Handbuch steht gar nichts darüber. Also: Cursorsteuerung Null, Bildschirmeditor: nicht vorhanden.

Bleiben wir noch beim Bildschirm: Der Textmodus erlaubt 16 Zeilen zu je 32 Zeichen; 13

verschiedene Grafikmodi sind möglich. Die Auflösung, der Speicherbedarf und die Farbvielfalt in den verschiedenen Modi sind unterschiedlich. Generell gilt, je höher die Auflösung, desto größer der Speicherbedarf und desto weniger Farben. Für das Arbeiten im Hochauflösungs-Modus ist daher ein Arbeitsspeicher von 16 K Bedingung, die Grundversion reicht, wie auch beim VC-20, nicht aus.

Als Herz des neuen Tandy schlägt ein 6809, betrieben mit cirka 800 kHz. Damit stehen Maschinenspracheprogrammierer, die hier 6502- oder Z-80-Kompatibles erwartet hatten, ebenfalls vor neuen Aufgaben. Das RAM ist dynamisch, und die anderen Support- und Peripheriechips entstammen ebenfalls der 6800-Familie. Wobei in Sachen Peripherie noch ein Tip angezeigt ist: Wenn Sie einen normalen Kassettenrecorder verwenden und Ihnen bricht beim 'Saven' der Pegel zusammen, dann hat Ihr Recorder einen kapazitv abgeblockten Eingang. Das aber mag der Colour Computer offenbar gar nicht. 10 kOhm gegen Masse lösen das Problem; der Pegel steht. In bezug auf die Belegung der DIN-Buchse für den Recorder sollten Sie wissen: Es wird nur der rechte Kanal benutzt. Mit einem Mono-Recorder tut sich ohne Adapterlöten trotz 'Normbuchse' nichts. Nachdem die Japaner das Prinzip der Belegung nach DIN bereits erfolgreich anwenden, bleibt zu hoffen, daß auch die Amerikaner eines Tages dazu in der Lage sein werden.

Fazit

Um einen Champion-Titel braucht sich Tandys Colour Computer 2 nicht zu bewerben — hier wird er von den zahlreichen Mitbewerbern sowohl preis- als auch leistungsmäßig sang- und klanglos in die Seile geschickt. Dagegen dürfte er ein guter Sparringspartner für denjenigen sein, der über fertige Software und Module einen Eingang in die Welt der kleinen Computer sucht.

Rechner	Programm							
	1	2	3	4	5	6	7	8
Apple II Plus	1,4	8,4	15,8	17,6	19,0	28,4	45,0	10,4
VC-20	1,2	8,1	15,3	16,8	18,1	27,1	43,0	9,6
TI-99/4A	3,4	8,5	24,5	25,0	26,5	62,2	84,8	38,9
CBM 3001	1,4	9,6	18,0	20,0	21,7	32,2	50,6	11,6
Dragon	1,2	8,6	17,0	18,0	19,5	28,9	42,3	10,9
ORIC-1	2,3	17,8	29,7	32,0	39,2	53,2	79,2	12,7
C-64	1,25	9,4	18,2	20,5	21,4	32,1	51,1	11,3
alphaTronic PC	2,2	5,3	15,4	16,7	18,1	31,0	42,6	17,8
BBC-ACORN	0,66	2,89	7,95	8,41	8,84	13,53	20,95	4,80
TRS-80 Colour 2	1,6	12,1	19,5	21,8	22,6	32,8	47,7	*

* Operator '↑' wurde nicht akzeptiert

Ergebnisse des Benchmark-Tests (Zeiten in Sekunden)

Ergebnisse auf einen Blick:

- ⊕ solide verarbeitet
- ⊕ Grafik durch Befehlssatz unterstützt
- ⊖ fehlender Bildschirmeditor
- ⊖ kein Monitorausgang
- ⊖ Standard-Joysticks wegen speziellen Anschlusses nicht verwendbar

te-wi aktuell...

Drei Titel, die Ihnen beim „Computern" helfen

CP/M und WordStar (Paul/Riedel)
Dieses Buch bietet dem ständig steigenden Kreis von Mikrocomputer-Anwendern eine fundamentale Einarbeitungshilfe. 144 Seiten, Softcover, DM 29,80*

MIKROCOMPUTER-GRUNDWISSEN (Adam Osborne)
Eine allgemeinverständliche Einführung in die Mikrocomputer-Technik – optimal als Einstieg für Elektronik-Laien. 304 Seiten, Softcover, DM 36,-*

6502 – Programmieren in Assembler (L. Leventhal)
Dieser Titel aus einer ganzen Reihe von Büchern über die Assemblersprachen-Programmierung befaßt sich ausführlich mit dem weitverbreiteten Mikroprozessor 6502. 704 Seiten, Softcover, DM 59,-*

* Die Preise sind die Ladenpreise.

tm 3814

te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40
te-wi

EBECO
Gert König
Computersysteme

Video-Genie

COLOR-Genie, 16 K	650.– DM
Video-Genie I/II, 64 K	1150.– DM

Zubehör TRS-80/Video-Genie

48-K-Speichererweit.	150.– DM
Expander: Uhr, Doubler, Druckerinterface, o. RAM	650.– DM
Bausatz ohne Doubler	330.– DM
Leerkarte + Unterlagen	90.– DM
Doubler TRS-80/Genie	250.– DM
Grafik 384 × 192, 12 K RAM	360.– DM
Bausatz komplett	250.– DM

Floppystationen

kpl. anschlußfertig: Doubler, Uhr, Kabel, Druckerinterf., TEAK-Laufwerke, NEWDOS80/2, Handbuch:

1 TEAK F55A, 170 KB	1050.– DM
2 TEAK F55B, 730 KB	1890.– DM
2 TEAK F55F, 1400 KB	2550.– DM
Aufpreis f. Controller	400.– DM

Laufwerke

TEAK F55A, 40 Spur., 250 KB, SS/DD	640.– DM
TEAK F55B, 40 Spur., 500 KB, DS/DD	750.– DM
TEAK F55E, 80 Spur., 500 KB, SS/DD	740.– DM
TEAK F55F, 80 Spur., 1,0 MB, DS/DD	880.– DM
TEAK F55G, 80 Spur., 1,6 MB, DS/DD	1040.– DM

Monitore

ZENITH ZVM 122, bernstein	330.– DM
TAXAN vision III RGB-TTL	1590.– DM

Andere Monitore auf Anfrage.

Disketten

Alle Disketten mit Verstärkungsring, 5 Jahre Garantie.
1. Preis 10 Stück – 2. Preis 100 Stück

Verbatim, 5", APPLE, SS/SD	5.90/5.30 DM
Verbatim, 5", VEREX, SS/SD	6.20/5.60 DM
Verbatim, 5", Datalife	7.70/6.92 DM
Verbatim, 8", VEREX, SS/SD	6.25/5.35 DM
SKYTEK, 5", SS/SD	5.47/5.35 DM
SKYTEK, 5", SS/DD	5.81/5.36 DM
SKYTEK, 5", DS/DD/80 Tr.	6.73/6.04 DM

IBM-PC

GENIE 16: 1:1, IBM-kompatibel in Hard- und Software, 8086-Prozessor (vorbereitet für 8087), 4,7-MHz-CPU-Takt, 208-K-Speicher, Farbgrafik 320 × 200 oder 640 × 200, 16 Farben, Anschluß f. SW- u. RGB-Monitor, Centronics + RS232, 4 IBM-Slots, 2 echte 16-Bit-Slots, 2 TEAC-F55B-Laufwerke je 360 KByte, inkl. MS/DOS, AT/DOS, BASIC und Assembler, Zusatzkarten für IBM-PC laufen ohne Änderung, Tastatur wie IBM

kpl. anschlußfertig, jedoch ohne Monitor	5900.– DM
Monitor mit grüner Röhre	720.– DM
Monitor TAXAN vision III RGB, 16 Farben	1590.– DM
Speichererweiterung für IBM-PC u. GENIE 16, 64 KByte + Parity-Bit	200.– DM

Reparaturen

für IBM-PC, TRS-80, Video-Genie u. a. schnell und preiswert durch eigene Werkstatt. Kostenvoranschläge innerhalb eines Tages. Nachfragen lohnt sich!

Alle Preise inkl. MwSt. ab Lager Köln. Änderung und Irrtum vorbehalten.

Ebeco G. König, St.-Anno-Str. 6, 5000 Köln 90, Tel. 0 22 03/2 83 87

c't-Club

— das ist Ihr Forum.
Wir veröffentlichen kostenlos Kontaktanzeigen von c't-Lesern, Nachrichten und Anschriften von Computer-Clubs. Schicken Sie einfach eine Postkarte an die **Redaktion c't.**
c't-Club, Postfach 27 46, 3000 Hannover 1

Das Club-Portrait

Die FORTH INTEREST GROUP DEUTSCHLAND sandte uns die foldende Selbstdarstellung:

FORTH ist ein außergewöhnliches Programmiersystem. Da es aber nicht von Großkonzernen oder Universitäten gefördert wird, ist seine Verbreitung hier in Deutschland noch recht gering. Um das zu ändern, hat sich die FORTH INTEREST GROUP DEUTSCHLAND gebildet. Diese Gemeinschaft ist etwas ganz anderes als eine Users Group. Bei FORTH kann der einzelne Benutzer selber seine Kreativität und seinen Einfallsreichtum in der Gestaltung der Sprache einbringen, so wie es in keiner anderen Sprache möglich ist.

Die FORTH INTEREST GROUP entstand anfänglich in Kalifornien, verbreitete sich auf die USA und teilweise nach Europa. Sie ist ein Phänomen ganz besonderer Art in der Welt der Microcomputer. Während Microcomputersoftware das Millionengeschäft unserer Zeit ist, hat sich um FORTH eine Gemeinschaft gebildet, die public domain software vertreibt. Hier ist Software so billig, daß sich Klauen nicht lohnt. Der Quellcode der Programme wird mitgeliefert. Während überall alles nach Konzentration strebt und möglichst als Aktiengesellschaft an die Börse geht, hier eine Vereinigung von kleinen Firmen und einzelnen Programmierern.

Eine Gemeinschaft von Spinnern? Weltfremde Computerbastler, die die Hoffnung aufgegeben haben, auch noch reich zu werden? Nun, FORTH ist eine der am meisten verwendeten Sprachen in Videospielen, nicht gerade der links liegengelassene Zweig in den letzten Jahren der Computerei. FORTH ist besonders stark in Prozess-Automaten und Robotics. Auch nicht gerade ein Aschenbrödel unter den Computeranwendungen.

Die Ziele der FORTH INTEREST GROUP Deutschland:

— Verbreitung des Forth-Konzepts,
— Verbreitung von billigen FORTH-Systemen für alle Microprozessoren,
— Verbreitung der FORTH-Literatur und
— Verbreitung von FORTH in Anwendungen.

Darüber hinaus besteht noch ein spezielles Ziel für den deutschen Sprachraum: Wir möchten FORTH von seiner Fixierung auf seine augenblickliche Syntax lösen, und eine deutsche Programmiersprache schaffen, die möglichst so natürlich ist, wie die geschriebene Sprache. Diese Sprache soll, nach dem ersten deutschen Computerwissenschaftler, 'LEIBNIZ' genannt werden.

Weiterhin ist es unsere Absicht, eine Computer-Cooperative ins Leben zu rufen, die 'Software Gilde Deutschland', deren Ziel es sein wird, Softwareherstellung im Rahmen von klein- und mittelständischen Betrieben zu ermöglichen, und somit den Zwang zu immer weiterer Konzentration auch dieser Branche zu vermindern.

Bei Interesse schreiben Sie bitte an:
FORTH INTEREST GROUP
c/o Andreas Goppold
Bethesdastraße 11C
2000 Hamburg 26
Telefon (0 40) 25 45 56

Bitte spezifizieren Sie, zu welchen Punkten Sie Informationen haben wollen. Legen Sie für die Rückantwort bitte einen selbstadressierten DIN A 5 Umschlag und DM 2,— in Briefmarken bei.
Das Treffen der FORTH INTEREST GROUP ist jeden vierten Samstag im Monat um 16 Uhr. Ort: hängt ab von der Zahl der Teilnehmer. Vorläufig: Common Interface Alpha, bei Lynsche, Holstenstr. 191, 2 Hamburg 50 (bitte vorher anrufen).

Club-Nachrichten und Adressen

PSI-80-Interessengruppe
Rupert Nieberle
Hermann-Föttinger-Institut
TU Berlin
Straße des 17. Juni 135
1000 Berlin 12

Sirius User-Club e. V.
Hamburg
Heinz von de Wauw
Ordulfstr. 72
2000 Hamburg 61
Programmtausch, Club-Infos 6 x im Jahr, Beitrag 60 Mark im Jahr, Programmierkurse, Einführungskurse

Hard- und Software-Club
für 6502, Z 80 (8086, Z 8000)
Rüdiger Höfert
Schöneberger Str. 72g
2000 Hamburg 73

SFC/PRC/CC Seneca
Dieter Eckhardt
Rückertstr. 1
4100 Duisburg 14

MC-Arbeitsgemeinschaft
SV Archigymnasium Soest
Michael Göder
Röttgersweg 1
4775 Lippetal 2

Der Club für c't 86 und IBM-kompatible Rechner sucht Anwender für Erfahrungs- und Programmaustausch.

Hermes Gehrke
Neusser Wall 54
5000 Köln 1
Telefon (02 21) 72 46 36

Sinclair-User-Club
Rolf Lumpe
Herrenheide
5138 Heinsberg

HP-41-Club Schwaben
Tom Schanz
Darmsheimer Str. 11
7032 Sindelfingen 6
Clubbeitrag 10 Mark pro Jahr, Clubzeitung geplant

Mikrocomputer-Interessentenkreis
Andreas Hilka
Seewiesenstr. 66
7343 Kuchen

Inter-Micro e. V.
W. Hebenbrock
Wesendonckstr. 19
8000 München 81
Tel. (0 89) 95 83 63

Neue Anschriften

KAYPRO-Club Nord:
c/o Dipl.-Ing. Giok L. Oey
Glinderweg 33
2050 Hamburg 80
Telefon (0 40) 7 24 81 72

KAYPRO-Club West:
c/o Dipl.-Ing. Jochen Hühne
Sedanstraße 4, **4134 Rheinberg**
Telefon (0 28 43) 7 33 78

CCK
(Computer-Club Karlsruhe)
Kontaktadresse:
Georg Lange, Badener Str. 7
7575 Ebersteinburg

Club 80
Kontaktadresse:
Günther Wagner
Gartenstraße 4
8201 Neubeuern
Tel. 0 80 35/33 61, (nur abends!)

Der neugegründete Unabhängige Computerclub Weiden bietet Hilfe bei Hard- und Softwareproblemen. Ein weiterer Schwerpunkt liegt auf dem Erfahrungsaustausch. Der Club unterstützt alle Rechnersysteme. Es finden monatliche Treffen statt.

Berthold Weber
Landgerichtsstraße 11
8480 Weiden
Telefon (09 61) 3 23 05

Schweiz

Sorcerer-User-Gruppe Schweiz
K. v. Grüningen
Jetzikofenstr. 14
CH-3038 Kirchlindach

Sorcerer-Gruppe
Werner Gribi
Postfach 9
CH-3294 Büren a. A.

Sharp CCS
J. S. Grabenstetter
Marsstr. 2
CH-4123 Allschwil

Erster bayerischer Computertag

Am 13. Mai findet im Löwenbräukeller in München der 1. bayerische Computertag statt. Der Eintrittspreis beträgt 5,— DM für Erwachsene und 3,— DM für Schüler und Studenten.

Informationen:
CTR Computer-Tage Redmann
Postfach 11 54
6073 Egelsbach

SE4941

Universeller EPROM-EEPROM-Programmer

Schnell und sicher können Sie mit dem SE 4941 im Labor oder "vor Ort" die meistverwendeten EPROM- und CMOS-Typen programmieren:

- hohe Betriebssicherheit durch vielseitige Testfunktionen (Selbsttests)
- V24-Schnittstelle
- PROM-Simulation (Option)
- 7 Datentransferformate
- Teil- und Mischfunktionen für 16-Bit-Anwendungen
- Editier- und Suchfunktionen für Änderungen vor Ort
- automatisches Programmieren vereinfacht die Bedienung
- kurze Programmierzeit durch neuen Programmier-Algorithmus
- Der Preis: 2.950 DM + MWSt
- Bei Nichtgefallen Rückgaberecht innerhalb von 2 Wochen

SE 3062 Bückeburg, Postfach 1308, Tel. 05722/203106
Teletex 572210, Telex 17572210

68008 ECB CPM/68K

- Single-Euro-Karte mit ECB-Bus
- 2 RS232 Vollduplex-Schnittstellen
- 2 Bytewide Sockel für max. 64 kB
- Power fail detect
- 16 kB Monitor: stand-alone lauffähig
- Wartungsfeld kann angeschlossen werden
- Wait synchronisation
- Bustiming: MREQ einstellbar 300 ns – 1 µs
- Interr-Level: NMI, INT, Real-time-clk
- Adressen A0 – A19 ohne Banking
- Refresh-Generierung (ohne Adressen)
- Single 5V-Versorgung
- gepuffertes Uhrenchip
- Baugruppe: DM 1.699 inkl. MWST
- kompl. Rechner: DM 14.799 inkl. MWST
- CPM/68 K für IHREN FD-Controller a.A.

dr. neuhaus

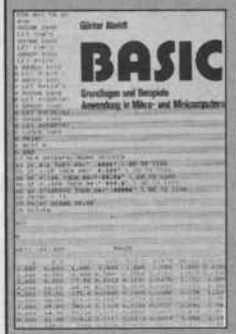
Hochallee 39 2000 Hamburg 13
Tel.: 040 / 44 17 02 Tx: 217 35 13 drn d

TOPP aktuell

Best.-Nr. 354
D. Böhm
Ich möchte einen Computer
DM 10,80

Best.-Nr. 355
D. Böhm
Computergesteuerte Meßtechnik
DM 25,80

Best.-Nr. 428
J. Kwiatkowski
FORTRAN in 8 Lektionen für Anfänger
DM 29,80



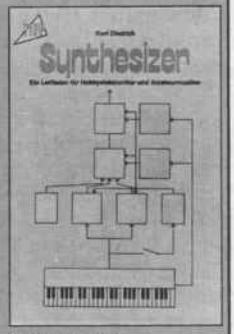
Best.-Nr. 455
G. Abeldt
BASIC – Grundlagen und Beispiele
DM 9,–

Best.-Nr. 401
Rowley
Atari BASIC
DM 10,80

Best.-Nr. 496
Libes/Wahl
Personal Computer Handbuch
DM 19,80

Best.-Nr. 475
M. Reinhard
Mikroprozessor SCMP II
DM 19,80

Best.-Nr. 420
L. Schüssler
Fernsteuern – Sonder- u. Zusatzfunktionen
DM 19,80



Best.-Nr. 424
K. Diedrich
Synthesizer
DM 14,80

Diese informative Buchreihe von TOPP wird ständig durch weitere Titel ergänzt, die zum Verständnis unserer Tage beitragen.

Prospekte über die Gebiete Elektronik und Amateurfunk stehen kostenlos zur Verfügung. – Bitte anfordern.

frech-verlag
Turbinenstr. 7, 7000 Stuttgart 31
Telefon (0711) *83 20 61, Telex 7 252 156 fr d

Wenn 64K nicht reichen

Banking, Paging, virtueller Speicher

Peter Glasmacher

Mikroprozessoren mit einer Wortbreite von 8 Bit, selbst so moderne Maschinen wie der 6809, stammen aus einer Zeit, in der die Entwickler solcher Chips der Meinung waren, ein Adreßbereich von 64KByte würde vollkommen ausreichen. Eine durchaus verständliche Annahme, erreichten doch selbst Größtrechner in den sechziger Jahren kaum diese Speicherkapazität. Immer aufwendigere Betriebssysteme und Anwenderprogramme im Mikrorechnerbereich verlangen aber nach immer mehr Halbleiterspeichern, um eine schnelle und komfortable Programmausführung zu ermöglichen. Der Preisverfall bei Speicher-ICs läßt nunmehr auch kostengünstige Lösungen zu. Nur die Beschränkung, die einem die geringe Anzahl Prozessorbeinchen (sprich: Adreßleitungen) auferlegt, gilt es zu beseitigen.

Die Speicherbelegung eines beliebigen (hypothetischen) Mikrocomputers zeigt Bild 1. (Das Layout ähnelt verblüffend der Speicheraufteilung der APPLE-Rechner, was daran liegt, daß der Autor sich gerade intensiv damit beschäftigt hat. Wie man sieht, bleibt für Anwenderprogramme bei weitem nicht soviel Speicher übrig, wie manche Angaben in Werbeprospekten vermuten lassen. Systemprogramme, wie Betriebssystem und Sprachinterpreter, erreichen auf modernen Tischrechnern ohne Anstrengung einen Umfang von 24 bis 32 KByte. Oft genug sind die beiden Komponenten auch noch in ROMs oder EPROMs untergebracht, so daß sie nur sehr schwer durch andere Programme zu ersetzen sind. Damit der verbleibende Adreßraum besser ausgenutzt, beziehungsweise scheinbar erweitert wird, kommen unter anderem die folgenden Methoden zum Einsatz:

1. Der gesamte verbleibende Speicher wird in einzelne Seiten aufgeteilt, ein spezieller Betriebssystemteil verwaltet Seiten und Programm(teile).
2. Programme werden ausschließlich als Quellcode auf einem Datenträger aufbewahrt und bei Bedarf compiliert und ausgeführt.
3. Einem kleinen Teil des adressierbaren Speichers, meist dem residenten Teil des Betriebssystems, wird durch eine einfache Hardwareschaltung ein gleich großer Schreib-Lesespeicher parallelgeschaltet.
4. Der Rechner besitzt eine Hauptspeicherkapazität, die einem Vielfachen des adressierbaren Speichers entspricht. Dieser Hauptspeicher ist in gleich große Teile (engl. 'BANKS') aufgeteilt, eine Kombination von Hard- und Software sorgt für ein reibungsloses Zusammenspiel.

Die beiden ersten Vorschläge lassen sich ausschließlich durch

Software-intensiv

geschicktes Programmieren verwirklichen. Ein schneller externer Massenspeicher (Diskettenstation oder Festplatte) ist dabei jedoch nötig, ein Kassettenrecorder wäre zu langsam. Fall 1 ist als virtuelles Speicherkonzept bekannt (Bild 2) und wird eigentlich nur auf Großrechnern angewandt, wo das Verhältnis vom vorhandenen internen Halbleiterspeicher zum adressierbaren Speicher auch heute noch oft zu Ungunsten des ersteren ausfällt. Da der Begriff 'virtueller Speicher' immer öfter auftaucht, soll das Prinzip kurz erklärt werden.

Der gesamte freie Speicherbereich, also alles, was nach Abzug des Betriebssystems übrigbleibt, wird in gleiche Segmente aufgeteilt. Diese Segmente set-

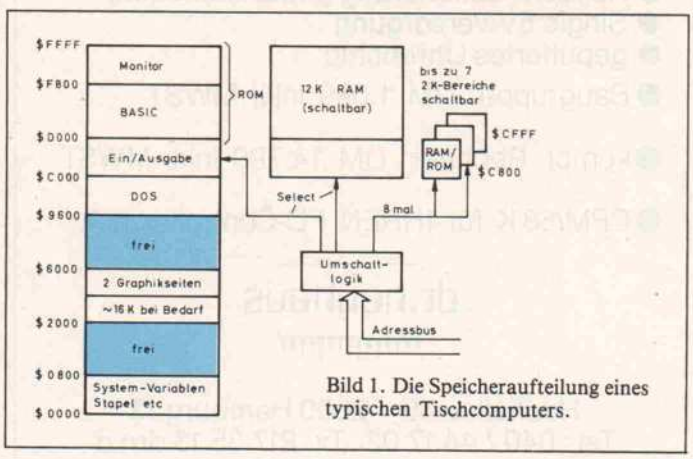


Bild 1. Die Speicheraufteilung eines typischen Tischcomputers.

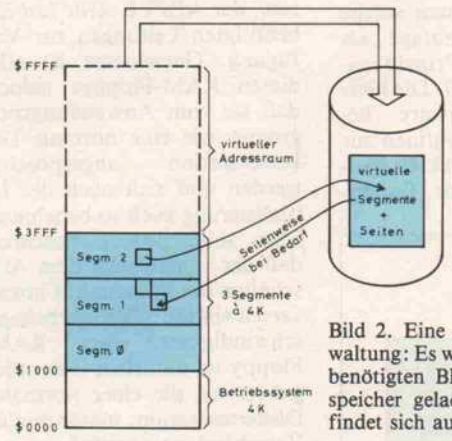


Bild 2. Eine virtuelle Speicherverwaltung: Es werden nur die zur Zeit benötigten Blöcke in den Primärspeicher geladen, alles andere befindet sich auf einer Magnetplatte.

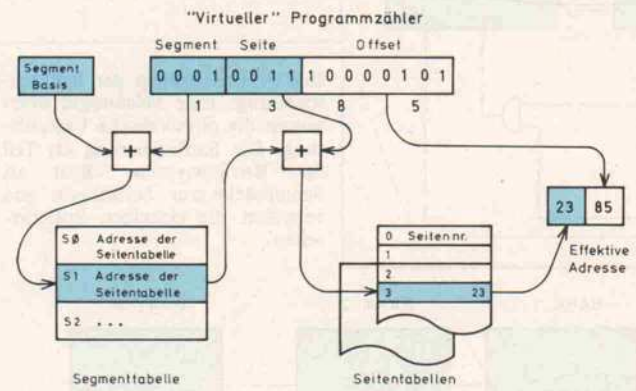


Bild 3. Umsetzung einer 'virtuellen' Adresse in eine effektive Speicheradresse. Durch Segment und Seitentabelle wird die physikalische Lage der Speicherstelle variabel.

zen sich wiederum aus kleineren Einheiten zusammen, die Seiten (Pages) genannt werden. Auf einem Plattenspeicher befindet sich nun ein 1:1-Abbild des gesamten adressierbaren Speichers (abzüglich des Betriebssystems). In den realen Speicher werden nur diejenigen Seiten und Programmteile geladen, die gerade ausgeführt werden, alles andere befindet sich auf der Platte. Damit auf Seiten und Segmente zugegriffen werden kann, enthält das Betriebssystem eine Segmenttabelle und für jedes Segment eine Seitentabelle (Bild 3).

Der Programmzähler zeigt nicht auf eine absolute Adresse wie im 'normalen' Betrieb, sondern besteht aus einem Segment- und Seitenindex sowie einem Offset innerhalb der Seite. Mit diesen Informationen kann das System auf jedes Byte innerhalb einer beliebigen Seite zugreifen. In der Seitentabelle ist für jeden Eintrag noch eine Markierung vorgesehen, die den Status der Seite, belegt oder frei, widerspiegelt. Mit all diesen Informationen verwaltet das jeweilige Betriebssystem den gesamten Speicher, und für den Außenstehenden wird so ein riesiger Hauptspeicher simuliert.

Wesentlich einfacher ist der zweite Fall. Hier befinden sich alle ausführbaren Programme als Quellcode auf einer Plattendatei. Ein Programm wird bei Bedarf übersetzt (compiliert) und gestartet. Werden nun weitere Programmteile benötigt, bietet das Betriebssystem (oder die Programmiersprache) die Möglichkeit, das im Speicher befindliche Programm ganz oder teilweise zu 'vergessen' und damit Platz für neue Programmteile zu schaffen. Die Vorteile dieses Systems liegen auf der Hand: Über ein einfaches Software-Interface (eine angenäherte Übersetzung könnte 'Schnittstelle zwischen Programmen' lauten) in Verbindung mit einem relativ kleinen Puffer (typisch 2K) werden unter Umständen riesige Programme ausgeführt (Bild 4). In einen solchen Blockpuffer lassen sich auch reine Datenblöcke laden, manipulieren und weitgehend automatisch auf die Datenträger zurückschreiben. Der nötige Programmieraufwand hält sich in Grenzen, Datenbestände sind gegen plötzliche Verluste, zum Beispiel durch Stromausfall, geschützt und der Hauptspeicher steht zum großen Teil für ausführbare Programme zur Verfügung (Bild 5).

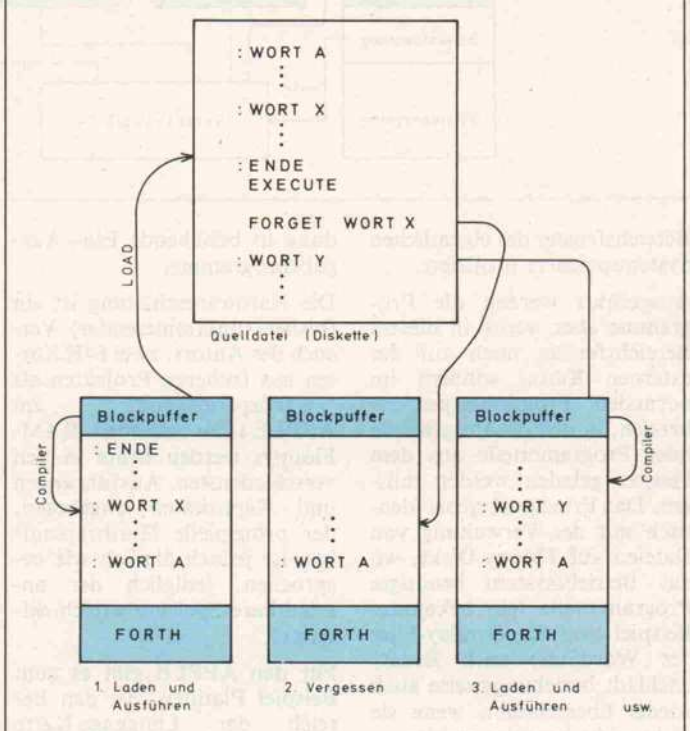


Bild 4. Die Ausführung von Programmen in FORTH. Das Wort FORTH an den richtigen Stellen ermöglicht die Ausführung von umfangreichen Programmen in einem vergleichsweise kleinen Speicher.

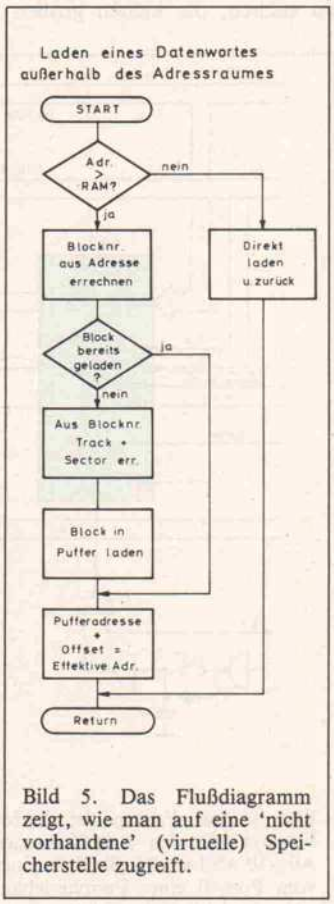


Bild 5. Das Flußdiagramm zeigt, wie man auf eine 'nicht vorhandene' (virtuelle) Speicherstelle zugreift.

Mit viel Silizium

Die bisher behandelten Software-Lösungen haben einen unübersehbaren Nachteil. Im ersten Fall muß ein umfangreiches Betriebssystem auf dem Rechner installiert sein, und außerdem entfaltet ein 'echtes' virtuelles Speicherkonzept seine Vorzüge erst in Verbindung mit Multitasking-Anwendungen. Multitasking ist auch einer dieser unübersetzbaren Fachausdrücke (wie Gemütlichkeit und Kindergarten) und bezeichnet den 'quasi gleichzeitigen' Ablauf mehrerer Programme auf einem Computersystem. 'Gleichzeitig' bedeutet aber meist, daß Programm B die Kontrolle erhält (und weiterarbeitet), während Programm A auf Daten von einer Ein-/Ausgabeeinheit wartet.

Der zweite Fall, obwohl wesentlich einfacher, verlangt eine Programmiersprache (zum Bei-

spiel FORTH), die konsequent für den Gebrauch eines Plattenspeichers entwickelt wurde. Leider erwachen Sprachen wie FORTH erst jetzt aus ihrem Dornröschenschlaf.

Es lag also nahe, eine Lösung zu suchen, die keinen großen Namen eingebracht: Im allgemeinen bezeichnet man solche Anordnungen heutzutage als RAM-Floppy. Das Prinzip zeigen die Bilder 6 und 7. Die hierzu benötigte Software beschränkt sich auf Routinen zur Errechnung der korrekten Fensterposition und zur Einbindung in bestehende Ein-/Ausgabeprogramme.

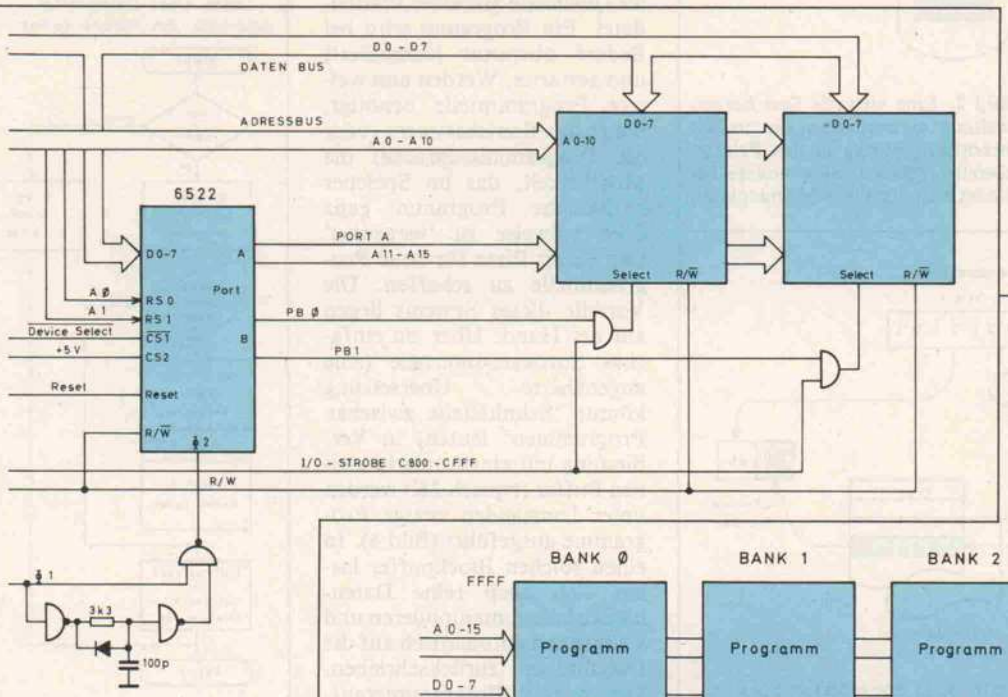


Bild 6. Das Prinzip der RAM-Floppy: Mit den Adreßleitungen A0—10 wird ein 2K-Block in einer vom Port B eines Peripheriebausteins selektierten Karte ausgewählt. Der Port A des Bausteins liefert die Adreßbits 11 bis 15.

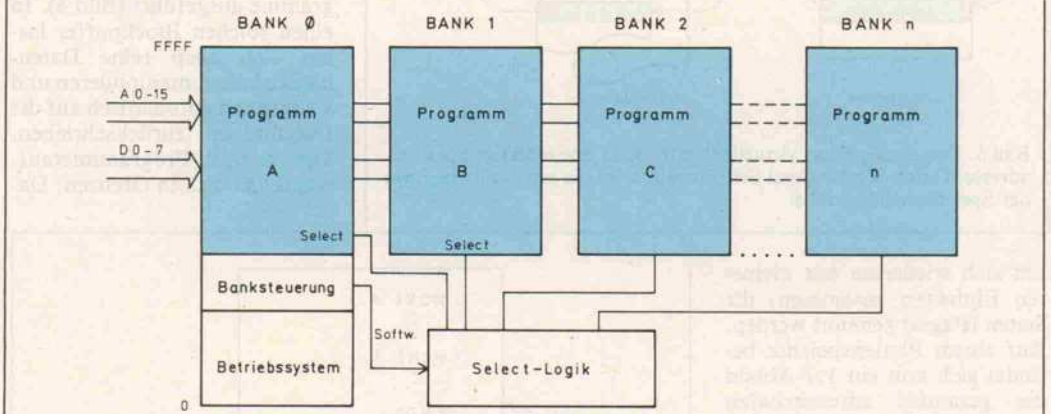


Bild 7. Das Prinzip der Bankumschaltung: Eine Selektlogik übernimmt die physikalische Umschaltung. Die Banksteuerung als Teil des Betriebssystems dient als Schnittstelle zur Selektlogik und verwaltet die einzelnen Speicherseiten.

Prinzip 'Nadelöhr'

Programmieraufwand erfordert. Zum Glück besitzen viele Mikrocomputer irgendwo innerhalb ihres Adreßraumes einen kleinen Bereich (2 bis 4K), der nicht genutzt wird (AIM 65) oder zur Aufnahme von Treiberprogrammen für Peripheriebausteine reserviert ist (APPLE). Mittels einiger Chips und etwas Denkarbeit läßt sich nun in diesem Bereich ein sehr umfangreicher Speicher unterbringen, indem man diesen (beispielsweise 2K großen) Ausschnitt wie ein Fenster über den Zusatzspeicher schiebt.

Noch einmal andersherum: Im direkt adressierbaren Bereich einer 8-Bit-CPU wird ein nicht benutzter Speicherbereich 'ausgeguckt'. In diesen werden Speicherinhalte eines zusätzlichen RAM-Systems eingeblendet. Wenn auf einer 'externen' 64K-RAM-Karte ein ausführbares Programm liegt, werden immer die aktuellen, also gerade benötigten Teilbereiche des Programmes in das reservierte Bereichsfenster des eigentlichen Systemspeichers projiziert.

Ausgeführt werden die Programme aber weder in diesem Bereichsfenster noch auf der externen Karte, sondern im normalen Programmspeicherbereich, in den die Programme oder Programmteile aus dem Fenster geladen werden müssen. Das Prinzip ist quasi identisch mit der Verwaltung von Dateien auf Floppy-Disks, wo das Betriebssystem benötigte Programmteile (ein bekanntes Beispiel sind die Overlay-Files des WordStar) nach Bedarf nachlädt beziehungsweise auch wieder überschreibt, wenn sie nicht mehr benötigt werden.

Die Ausrichtung auf diese Form der Ein-/Ausgabe hat dem Kind letztlich auch seinen Adreßbereiches etwas anders sein, der APPLE stellt fast alle benötigten Leitungen zur Verfügung. Gemeinsam ist allen diesen RAM-Floppys jedoch, daß sie vom Anwendungsprogramm wie eine normale Diskettenstation angesprochen werden und sich nach der Initialisierung auch so benehmen. Man sollte jedoch beachten, daß der Inhalt nach dem Ausschalten des Systems auf immer verschwindet. Die Arbeitsgeschwindigkeit einer RAM-Floppy ist natürlich wesentlich größer als die einer normalen Diskettenstation; mechanischer Verschleiß ist ebenfalls ausgeschlossen.

Die Hardwareschaltung ist ein (bisher funktionierender) Versuch des Autors, zwei 64K Karten aus früheren Projekten als 'Festkörperdiskette' am APPLE II zu betreiben. RAM-Floppys werden heute in den verschiedensten Ausführungen und Kapazitäten angeboten, der prinzipielle Hardwareaufbau ist jedoch ähnlich wie besprochen, lediglich der anschaltbare Speicherbereich differiert.

Für den APPLE gibt es zum Beispiel Platinen, die den Bereich der Language-Karte (12K) als Fenster benutzen. Bei anderen Tischcomputern dürfte die Erzeugung der diversen Signale zur Selektierung des

Elegant, aber aufwendig

Eine weitere zu besprechende Variante der Speichererweiterung ist die Umschaltung von vergleichsweise großen Speicherblöcken, allgemein unter dem Begriff 'Banking' oder 'Bankswitching' bekannt. Darunter versteht man das Anschalten eines Speicherblocks (einer 'Bank' also), dessen Größe oft dem gesamten adressierbaren Bereich des verwendeten Prozessors entspricht. Die Umschaltung kompletter Adreßbereiche wirft nicht unerhebliche Probleme auf, die bei den bisher behandelten Lösungen nicht auftreten.

Jedermann kann sich vorstellen, was passiert, wenn auf ei-

nen Speicherbereich umgeschaltet wird, der neue Speicherblock aber alles mögliche, nur nicht die Fortsetzung des gerade abgearbeiteten Programms enthält. Die modernsten 8-Bit-Chips, zum Beispiel der Z800 (in Musterstückzahlen vielleicht noch in diesem Jahr erhältlich), sind für einen derartigen Fall gerüstet: Ein illegaler Befehl oder ein Zugriff auf einen unerlaubten Speicherbereich erzeugen einen speziellen Interrupt (Programm-Check). Eine entsprechende Unterbrechungsroutine kann dann zu einem kontrolliertern Programmabbruch, oder besser, zur Behebung des Fehlers verwendet werden.

Die derzeit erhältlichen Prozessoren sind für diesen Fall jedoch denkbar schlecht ausgestattet: Falsche Instruktionen führen sie üblicherweise in das 'never never land', wie es in den USA bezeichnet wird.

Derartige Probleme lassen sich auf verschiedene Arten umgehen. So kann man festlegen, daß nicht der gesamte Adreßbereich auf einen Schlag umgeschaltet wird, daß beispielsweise das Betriebssystem stets in alle Bankbereiche eingeblendet wird.

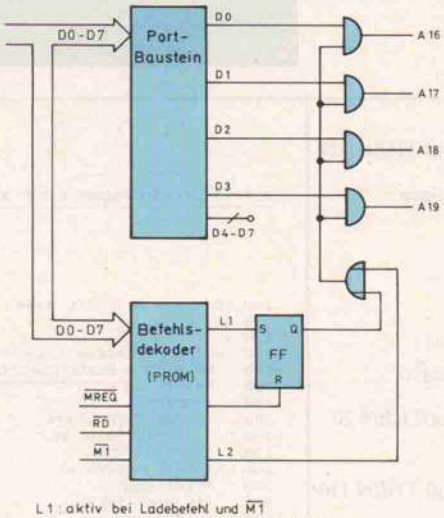
Will man aber den gesamten Adreßbereich umschalten, steht man vor folgendem Problem: Wie bekommt man denn Programmteile in andere Speicherbereiche, wo sie vorhanden sein müssen, **bevor** man auf diesen Speicherbereich überhaupt gefahrlos zugreifen darf? Ein Umkopiervorgang des Betriebssystems beziehungsweise eines auszuführenden Programmes könnte wie folgt aussehen: Durch geschickte Einbeziehung der Schreib-/Leseleitung des Prozessors kann man erreichen, daß bei jeder Leseoperation im aktuellen Speicherbereich gelesen wird, bei einer entsprechenden Schreiboperation aber in die Speicherbank geschrieben wird, auf die umgeschaltet werden soll. Auf die Art lassen sich sehr schnell komplette Speicherbereiche umkopieren. Nach erfolgreicher Übertragung kann gefahrlos in die andere Bank gewechselt werden.

Eine andere Methode benutzt bestimmte Prozessorbefehle zum Ansprechen einer anderen Bank. Parallel zum Prozessor-Datenbus liegt ein Dekoder (üblicherweise ein PROM), das nur beim Auftreten der vereinbarten Befehle aktiv wird. Die eigentliche Bankadresse wird auch hierbei üblicherweise mittels eines Portbausteines, über den ein geeignetes Datenwort ausgegeben wird, eingestellt.

Um nun zunächst wieder das Problem zu lösen, in die angewählte Bank Daten zu übergeben, ohne sie zu 'betreten', wählt man sich zwei Ladebefehle aus, beim Z80 zum Beispiel die Befehle LD A,(BC) beziehungsweise sein Gegenstück, LD (BC),A. Diese Befehle sollten sinnvollerweise Ein-Byte-Befehle sein (hier 0AH und 02H), um die Dekodierung nicht unnötig zu komplizieren.

Bild 8 zeigt die erforderliche Logikschaltung. Wenn während eines Opcode-Fetch (M1-Zyklus) einer der beiden Befehle auftritt, wird ein Flip-Flop (FF) gesetzt, das die Banking-Adressen am Ausgangsport aktiviert. Sind diese nicht gerade zu null gesetzt, wird nach dem M1-Zyklus, bei der Ausführung des Befehles, über BC eine Speicherzelle in der neu ausgewählten Bank adressiert. Jeder anderslautende Befehl setzt das Flip-Flop beim nächsten Auftreten von M1 wieder zurück. Das Prinzip einer RAM-Floppy, also die Datenübergabe zwischen Bankbereichen, ist damit bereits realisiert.

Um Programme in anderen Bereichen ablaufen zu lassen, muß die Bankadresse dauerhaft anliegen, also nicht nur für einen Befehl. Das kann mit anderen Ein-Byte-Befehlen erreicht werden. Diese sollten aber möglichst Sprungbefehle sein, weil man damit dann ein laufendes Programm an beliebiger (allerdings sinnvoller!) Stelle in der neuen Bank fortsetzen kann. Hier bieten sich beim Z80 JP(HL) oder bedingte RETurn-Befehle an. In Bild 8 wird davon ausgegangen, daß zum Umschalten und zum Zurückschalten auf Bank 0 (L2 inaktiv) verschiedene Befehle verwendet werden. Zumindest der Befehl zum Zurückschalten sollte nicht anderweitig im auszuführenden Programm auftreten, weshalb sich bedingte RETurns, beispielsweise auf Parity oder Vorzeichen, besonders eignen. Da allerdings die meisten Betriebssysteme aus einer Zeit stammen, als man um jedes Byte Speicherraum rang, steht zu befürchten, daß man darin schon derartige Befehle findet. JP (HL) bietet sich für CP/M daher eher an, da dieser Befehl im 8080-Befehlssatz, auf den sich die meisten CP/Ms beschränken, nicht enthalten ist.



L1: aktiv bei Ladebefehl und $\overline{M1}$
L2: aktiv nach vereinbartem Sprungbefehl (wird durch weiteren Befehl rückgängig gemacht)

Bild 8. Bankumschaltung mit 'reservierten' Prozessorbefehlen

Besondere Aufmerksamkeit sollte man bei allen Verfahren, die vollständig auf eine neue Bank wechseln, dem korrekten Transfer des Stacks widmen, ganz besonders, wenn man über RET-Befehle Bankwechsel vornimmt.

Auch die Hardware-Speichererweiterungen benötigen 'ihre' Software, um einwandfrei funktionieren zu können. Es liegt auf der Hand, daß im Betriebssystem Programmteile existieren müssen, die den Zustand des Speichers festhalten. Konkret gesagt, das Betriebssystem muß jederzeit wissen, welche Speicherseite momentan die Kontrolle bekommt und in welchem Zustand (aktiv/inaktiv) sich alle anderen Speicherseiten befinden. Durch die verwaltungstechnischen Probleme erhöht sich die Gefahr von unerwarteten Fehlern, wie oben bereits angeschnitten. Wenn man dann noch bedenkt, daß die Umschaltung von Speicherbereichen mit Vorliebe in Mehrbenutzersystemen angewendet wird, kann sich der Leser leicht vorstellen, welche Eventualitäten vom Entwickler beachtet werden müssen.

Wenn der adressierbare Speicherraum nicht mehr ausreicht, kann man dem durch eine Reihe verschiedener Verfahren abhelfen. Abgesehen vom virtuellen Speicherkonzept, welches nicht sehr oft im Tischrechnerbereich eingesetzt wird, muß man wohl von Fall zu Fall entscheiden, welcher Methode man den Vorzug gibt. Abhängig von konkreten Anforderungen kann jedes der angesprochenen Verfahren mit Erfolg eingesetzt werden. Anwender, die nicht gerade einen im Laufe der Jahre zum Jumbo ausgebauten Kim-1 besitzen, sondern sich mit einer 'Box' begnügen müssen, werden wohl einer der Softwarelösungen den Vorzug geben. RAM-Floppy und Bankumschaltung erfordern immer (teilweise drastische) Eingriffe in das Innenleben der Rechner.

Welche Lösung letztendlich sinnvoll und aufwandsmäßig vertretbar ist, wird man im Einzelfall entscheiden müssen. □

Literatur:
IBM/370 Principles of Operation
Form Nr. GA22-7000-B

Camp, Smay, Triska:
Microprozessor Systems Engineering
Matrix Publishers 1979

Glasmacher:
FORTH anpassen,
Ein Beispiel c't 1984

SBASIC-Prozessor

Strukturiert programmieren in Commodore-BASIC

Oliver Fischer

Der SBASIC-Prozessor besteht aus zwei Teilen und erlaubt auf CBM-Computern eine strukturierte Programmentwicklung. Dies bedeutet, daß einerseits Einrückungen im Programmtext möglich sind und andererseits Sprachelemente wie REPEAT-UNTIL, WHILE-DO, IF-THEN-ELSE, CALL PROCEDURE bei der Programmentwicklung verwendet werden dürfen. Das Programm ist dabei im wesentlichen ein Precompiler, der die neuen Sprachelemente in Commodore-BASIC übersetzt. Die Programme wurden auf einem CBM 3032 mit 4040-Floppy entwickelt und sind mit kleinen Veränderungen auch für andere Commodere-Rechner verwendbar. Eine Anpassung an Computer anderer Hersteller ist dagegen schwieriger. Allerdings läßt sich trotzdem das Grundkonzept des Precompiler übertragen.

Beim Erstellen von umfangreichen Programmen in BASIC kommt irgendwann der Zeitpunkt, an dem man aus dem selbsterstellten GOTO- und GOSUB-Wirrwarr nicht mehr herausfindet. Selbst wenn es dann gelungen ist, das Programm fertigzustellen, so bleibt doch die Unübersichtlichkeit von vielen GOTO- und GOSUB-Statements. Daran ändern auch noch so viele Kommentare im Programm nichts. Eine nachträgliche Änderung im Programm resultiert meist in einer Verfolgung von Sprüngen quer durchs Programm. Sicher kennt jeder die einzig mögliche Abhilfe, nämlich die strukturierte Programmierung. Doch wie soll man das verwirklichen, wenn die meisten Mikrocomputerhersteller in ihrem BASIC die notwendigen Sprachelemente für ein strukturiertes Programmieren einfach nicht vorsehen?

Im Commodore-BASIC sieht es in dieser Hinsicht besonders

WHILE-ENDWHILE

```
 10 !WHILE Bedingung !DO          10 IF NOT(Bedingung) THEN 110
 20   " " "                       20 " " "
 "    Beliebige Anweisungen   →   " Beliebige Anweisungen
 90   " " "                       90 " " "
100 !ENDWHILE                    100 GOTO 10
110 " " "                        110 " " "
```

REPEAT-UNTIL

```
 10 !REPEAT                       10 REM
 20   " " "                       20 " " "
 "    Beliebige Anweisungen   →   " Beliebige Anweisungen
 90   " " "                       90 " " "
100 !UNTIL Bedingung             100 IF NOT(Bedingung)THEN 20
```

IF-THEN-ELSE-ENDIF

```
 10 !IF Bedingung !THEN           10 IF NOT(Bedingung) THEN 110
 20   " " "                       20 " " "
 "    Beliebige Anweisungen       " Beliebige Anweisungen
 90   " " "                       90 " " "
100 !ELSE                       →100 GOTO 200
110 " " "                        110 " " "
""" Beliebige Anweisungen        """ Beliebige Anweisungen
190 " " "                        190 " " "
200 !ENDIF                       200 REM
```

CALL 'Name' — PROCEDURE 'Name' — ENDPROC

```
 10 !CALL 'Name'                  10 GOSUB 100
 20   " " "                       20 " " "
 "                            →   "
 80 " " "                         80 " " "
 90 END                           90 END
100 !PROCEDURE 'Name'           →100 REM
110 " " "                        110 " " "
"""                              """
140 " " "                        140 " " "
150 !ENDSPRUNG                   150 !RETURN
```

Tabelle 1. Gegenüberstellung 'SBASIC'-Commodore-BASIC

PROGRAMMBEISPIEL:

```
100 REM **** BEISPIEL ****
110 :
120 :
130 !WHILE LEFT$(EN$,1)<>"N" AND LEFT$(EN$,1)<>"J" !DO
140   PRINT"□  BERECHNUNG DES ARCCOS UND ARCSIN"
150   PRINT"                                        "
160   !REPEAT
170     INPUT"ZAHL";A$
180     !CALL "STR-VAL"
190     PRINT"□"
200   !UNTIL ABS(A)<=1
210   PRINT"   "
220   !IF A=0 !THEN
230     AC=π/2
240   !ELSE
250     AC=ATN(SQR(1-A*A)/A)
260     IF AC<0 THEN AC=π+AC:REM NORMALES IF-THEN !!
270     IF AC=0 AND SGN(A)=-1 THEN AC=π:REM NORMALES IF-THEN !!
280   !ENDIF
290   !IF A=1 !THEN
300     AS=π/2
310   !ELSE
320     !IF A=-1 !THEN
330       AS=-π/2
340     !ELSE
350       AS=ATN(A/SQR(1-A*A))
360     !ENDIF
370   !ENDIF
380   PRINT"ACOS(";A;")=";AC
390   PRINT"ASIN(";A;")=";AS
400   INPUT"   ENDE";EN$
410 !ENDWHILE
420 END
430 !PROCEDURE "STR-VAL"
440   A=VAL(A$)
450 !ENDPROC
460 END
READY.
```

Tabelle 2. So übersichtlich ist ein SBASIC-Programm

schlimm aus. Es beginnt schon einmal damit, daß zu Beginn einer BASIC-Zeile keinerlei Leerzeichen akzeptiert werden, also Einrückungen von vornherein ausgeschlossen sind. Es mag zwar Speicherplatz sparen und den Interpretationsvorgang geringfügig beschleunigen, wenn bei der Abspeicherung einer Zeile sämtliche Leerzeichen am Zeilenanfang entfernt werden, aber der Übersichtlichkeit eines Programms ist eine solche Vorgehensweise bestimmt nicht dienlich. Selbst der ELSE-Befehl, der mittlerweile schon in vielen BASIC-Dialekten implementiert ist, fehlt im Commodore-BASIC. Darüber hinaus wäre es auch wünschenswert, wenn man über Konstrukte wie WHILE-ENDWHILE, REPEAT-UNTIL, CALL 'Unterprogrammname', wie sie zum Beispiel in PASCAL gibt, verfügen könnte.

Die beiden nachfolgenden Programme beheben nun diese Mängel im Commodore-BASIC. Das erste Programm 'SPACES', ein Assemblerprogramm, greift in den Vorgang des Abspeicherns von BASIC-Zeilen ein und erlaubt führende Leerzeichen zu Beginn jeder BASIC-Zeile. Das zweite Programm 'SBASIC' ist in Commodore-BASIC geschrieben und arbeitet als Preprozessor. Das bedeutet, daß ein mit den oben erwähnten Sprachelementen erstelltes strukturiertes BASIC-Programm in ein äquivalentes, interpretier- und compilierbares Commodore-BASIC-Programm übersetzt wird. Mit diesen beiden Programmen ist es nun möglich, in BASIC zu programmieren und doch übersichtliche, strukturierte Programme zu erstellen. GOTO's sollten deshalb nur noch dann im Programm auftauchen, wenn beispielsweise Fehlerroutinen angesprungen werden.

Beide Programme dürften nach einigen Anpassungen auch auf anderen Commodore-Systemen verwendbar sein. Doch nun zur Beschreibung der Programme:

Eingriff in CGR

'SPACES' ($033A — $03AA) dient dazu, am Anfang einer BASIC-Zeile Einrückungen zu ermöglichen. Dieses Maschinenspracheprogramm, das einen Teil des zweiten Kassettenpuffers belegt, ist speziell auf CBM-Computer der 3000er-Serie zugeschnitten und daher nicht ohne weiteres auf andere Commodore-Rechner wie CBM 8032, VC 20,C 64 übertragbar. Die notwendigen Änderungen sind aber geringfügig, so daß jeder, der im Besitz einer Vergleichstabelle für Zero-Page-Belegung und ROM-Routinen ist, diese selbst vornehmen kann. Die nachfolgenden Erläuterungen zur Arbeitsweise des Programms werden dafür sicher hilfreich sein.

Die einzige Möglichkeit, in den Arbeitsprozeß des BASIC-Interpreters einzugreifen, besteht darin, die Character-Get-Routine (CGR), die sich glücklicherweise in der Zeropage befindet, zu verändern. Mit dieser Routine liest der Interpreter jedes Zeichen aus einem BASIC-Test, sei es nun beim Einfügen einer neuen Zeile oder bei der Ausführung eines Programms. Ein Teil der CGR wird nun beim erstmaligen Aufruf von 'SPACES' (SYS 826) derart verändert, daß die CGR zunächst verlassen wird (JMP $033D) und nach Abarbeitung des Restes von 'SPACES' der Rücksprung erfolgt. Diese Veränderung der CGR erfolgt reversibel, so daß ein erneuter Aufruf mit SYS 826 die ursprüngliche CGR wiederhergestellt ($033A-$034C).

Da die CGR von vielen Stellen des Interpreters angesprungen wird, die normale CGR aber nur dann ersetzt werden soll, wenn eine neue BASIC-Zeile übernommen oder eine alte Zeile verändert werden soll, ist es zunächst nötig, den Aufruf der CGR zurückzuverfolgen. Wurde die CGR vom Unterprogramm 'Festkommazahl von BASIC-Zeile holen' ($C8A9) und dieses Unterprogramm wiederum von der Interpreterroutine 'Neue BASIC-Zeilen handhaben'($C3AD) aufgerufen, so muß 'SPACES' eingreifen. Andernfalls wird die normale CGR fortgeführt ($0370-$0381).

Diese Rückverfolgung wird durch eine Überprüfung der letzten beiden Rücksprungadressen auf dem Stack durchgeführt ($034D-$036F). War die Überprüfung positiv, so wird die modifizierte CGR ausgeführt ($0384-$03A8). Diese Routine eliminiert nun nicht, wie die normale CGR, eventuelle Leerzeichen zwischen Zeilennummer und BASIC-Text, sondern stellt den Character-Get-Pointer auf das erste Space in der BASIC-Zeile und übergibt anschließend die Kontrolle wieder an den Interpreter. Der Interpreter bindet nun seinerseits die BASIC-Zeile nach Umwandlung der Befehlswörter in ihre Tokens in den betreffenden Programmtext ein. Noch ein Hinweis: Das Programm 'SPACES' kann

```
                        0100          ; PROGRAM 'SPACES'
                        0110          ;------------------
                        0120          ;(C) OLIVER FISCHER
                        0130          ;  DEZEMBER 1983
                        0140          ;------------------
                        0150          ;
                        0160 CHRGET   .DE #0079
                        0170 CHRGETPTR .DE #0077
                        0180 STACK    .DE #0101
                        0190 MASKE1   .DE #85
                        0200 MASKE2   .DE #77
                        0210 MASKE3   .DE #B3
                        0220 SPACE    .DE #20 '
                        0230 COLON    .DE #3A
                        0240          .BA #033A
                        0250          .MC #033A
                        0260          .OS
                        0270          ;
                        0280          ;CHRGET-ROUTINE VERANDERN
                        0290          ;      (JMP #033D)
                        0300          ;
033A- A5 79             0310          LDA *CHRGET
033C- 49 85             0320          EOR #MASKE1
033E- 85 79             0330          STA *CHRGET
0340- A5 7A             0340          LDA *CHRGET+1
0342- 49 77             0350          EOR #MASKE2
0344- 85 7A             0360          STA *CHRGET+1
0346- A5 7B             0370          LDA *CHRGET+2
0348- 49 B3             0380          EOR #MASKE3
034A- 85 7B             0390          STA *CHRGET+2
034C- 60                0400          RTS
                        0410          ;
                        0420          ;ADRESSE DES AUFRUFENDEN
                        0430          ;PROGRAMMTEILS FESTSTELLEN
                        0440          ;
034D- 8D A9 03          0450          STA HELPA
0350- 8E AA 03          0460          STX HELPX
0353- BA                0470          TSX
0354- BD 01 01          0480          LDA STACK,X
0357- C9 A9             0490          CMP #$A9
0359- D0 15             0500          BNE NORMGET
035B- BD 02 01          0510          LDA STACK+1,X
035E- C9 C8             0520          CMP #$C8
0360- D0 0E             0530          BNE NORMGET
0362- BD 03 01          0540          LDA STACK+2,X
0365- C9 AD             0550          CMP #$AD
0367- D0 07             0560          BNE NORMGET
0369- BD 04 01          0570          LDA STACK+3,X
036C- C9 C3             0580          CMP #$C3
036E- F0 14             0590          BEQ MODGET
                        0600          ;
                        0610          ;NORMALE CHRGET-ROUTINE FORTSETZEN
                        0620          ;
0370- AD A9 03          0630 NORMGET  LDA HELPA
0373- AE AA 03          0640          LDX HELPX
0376- C9 3A             0650          CMP #COLON
0378- B0 2E             0660          BCS RETURN2
037A- C9 20             0670          CMP #SPACE
037C- D0 03             0680          BNE NOSPACE
037E- 4C 70 00          0690          JMP $0070
0381- 4C 81 00          0700 NOSPACE  JMP $0081
                        0710          ;
                        0720          ;MODIFIZIERTE CHRGET-ROUTINE
                        0730          ;
0384- A0 FF             0740 MODGET   LDY #$FF
0386- C8                0750 LOOP     INY
0387- B1 77             0760          LDA (CHRGETPTR),Y
0389- C9 20             0770          CMP #SPACE
038B- F0 F9             0780          BEQ LOOP
038D- C9 3A             0790          CMP #COLON
038F- B0 10             0800          BCS RETURN1
0391- 38                0810          SEC
0392- E9 30             0820          SBC #$30
0394- 38                0830          SEC
0395- E9 D0             0840          SBC #$D0
0397- B0 08             0850          BCS RETURN1
0399- AA                0860          TAX
039A- 98                0870          TYA
039B- 65 77             0880          ADC *CHRGETPTR
039D- 85 77             0890          STA *CHRGETPTR
039F- 8A                0900          TXA
03A0- 60                0910          RTS
03A1- C0 00             0920 RETURN1  CPY #$00
03A3- F0 02             0930          BEQ M1
03A5- E6 77             0940          INC *CHRGETPTR
03A7- 38                0950 M1       SEC
03A8- 60                0960 RETURN2  RTS
                        0970          ;
                        0980          ;HILFSSPEICHER ZUR RETTUNG
                        0990          ;              VON AKKU,X-REGISTER
03A9-                   1000 HELPA    .DS 1
03AA-                   1010 HELPX    .DS 1
                        1020          .EN
```

```
--- LABEL FILE: ---

CHRGET =0079            CHRGETPTR =0077         COLON =003A
HELPA =03A9             HELPX =03AA             LOOP =0386
M1 =03A7                MASKE1 =0085            MASKE2 =0077
MASKE3 =00B3            MODGET =0384            NORMGET =0370
NOSPACE =0381           RETURN1 =03A1           RETURN2 =03A8
SPACE =0020             STACK =0101
```

nicht zusammen mit BASIC-Erweiterungen, wie z. B. Toolkit, Exbasic etc. zusammenarbeiten, da diese ebenfalls die CGR verändern.

'SPACES' ist hier als Assemblerlisting abgedruckt, Tabelle 3 zeigt den dazugehörigen BASIC-Loader.

Zweimal durch den Quellentext

'SBASIC' belegt etwa 8 KByte; dieses vollständig in BASIC geschriebene Programm erledigt die Umsetzung der neuen Sprachelemente in eine äquivalente Commodore-BASIC-Form. Dazu sind zwei Durchläufe (Passes) durch den Quelltext notwendig, da bei der Umsetzung einiger Sprachelemente Sprünge mit Vorwärtsbezügen auftreten, die erst im zweiten Durchlauf abgedeckt werden können. Das Programm benötigt unbedingt die Unterstützung durch ein Floppy-Disk-Laufwerk, da nur so ein BASIC-Programm wie eine normale sequentielle Datei Zeile für Zeile eingelesen werden kann.

Nach dem Start des Programms wird man zunächst aufgefordert, die Namen des Quell- und des Zielfiles sowie die dazugehörigen Laufwerknummern einzugeben. Anschließend eröffnet das Programm die Quell- und Zieldateien und durchläuft Pass 1, in dem die Hauptarbeit des Übersetzers geleistet wird.

Im zweiten Durchlauf werden dann die noch offengelassenen Sprungadressen ergänzt. Das Programm führt eine Syntaxprüfung der neuen Sprachkonstrukte durch und gibt entsprechende Fehlermeldungen auf dem Bildschirm aus. Außerdem wird die Zeilennummer der gerade bearbeiteten Zeile angezeigt. Nach zwei fehlerfreien Durchläufen befindet sich das in Commodore-BASIC übersetzte Programm unter dem eingegebenen Namen auf der Zieldisk und kann sofort mit RUN gestartet werden. Ebenso ist natürlich eine anschließende Compilierung möglich.

Einige allgemeine Einschränkungen, die das Quellprogramm betreffen:

— Jedes SBASIC-Programm muß als letzte Programmzeile eine END-Zeile haben (z. B. 10000 END).

— Jedes Befehlswort der SBASIC-Erweiterung muß mit einem Ausrufungszeichen ('!') beginnen. Dies erlaubt es dem Preprozessor, die neuen Befehlsworte schneller zu finden.

— In jeder Programmzeile darf nur ein neues Sprachelement auftreten. Es dürfen auch keine Befehle des normalen Commodore-BASIC nachfolgen. Dies bedeutet, daß ein neues Befehlswort als erstes in der Zeile stehen muß. Man sollte auch sonst möglichst nur einen BASIC-Befehl pro Zeile vorsehen. Ansonsten geht in den vielen Fällen die Struktur des Programms wieder verloren.

— Die Schachtelungstiefe von WHILE-ENDWHILE-, REPEAT-UNTIL-Konstrukten ist im Programm auf max. 31 festgelegt, kann aber durch eine größere Dimensionierung der Feldvariablen R1 und W1 leicht erhöht werden.

— In einem Programm dürfen je maximal 51 WHILE-ENDWHILE- und IF-THEN-ELSE-ENDIF-Elemente auftreten. Dies kann ebenfalls erhöht werden (Dimensionierung von W2, I1,I2 erhöhen).

— Ein Programm darf in der abgedruckten Version höchstens 51 Unterprogramme enthalten (Felder P1$[ ],P1[ ]).

Die genaue Syntax von SBASIC ist in Form von Syntaxdiagrammen unten dargestellt. Tabelle 1 zeigt, wie die neuen Sprachelemente in ihr Commodore-BASIC-Äquivalent übersetzt werden.

Noch zwei Hinweise:

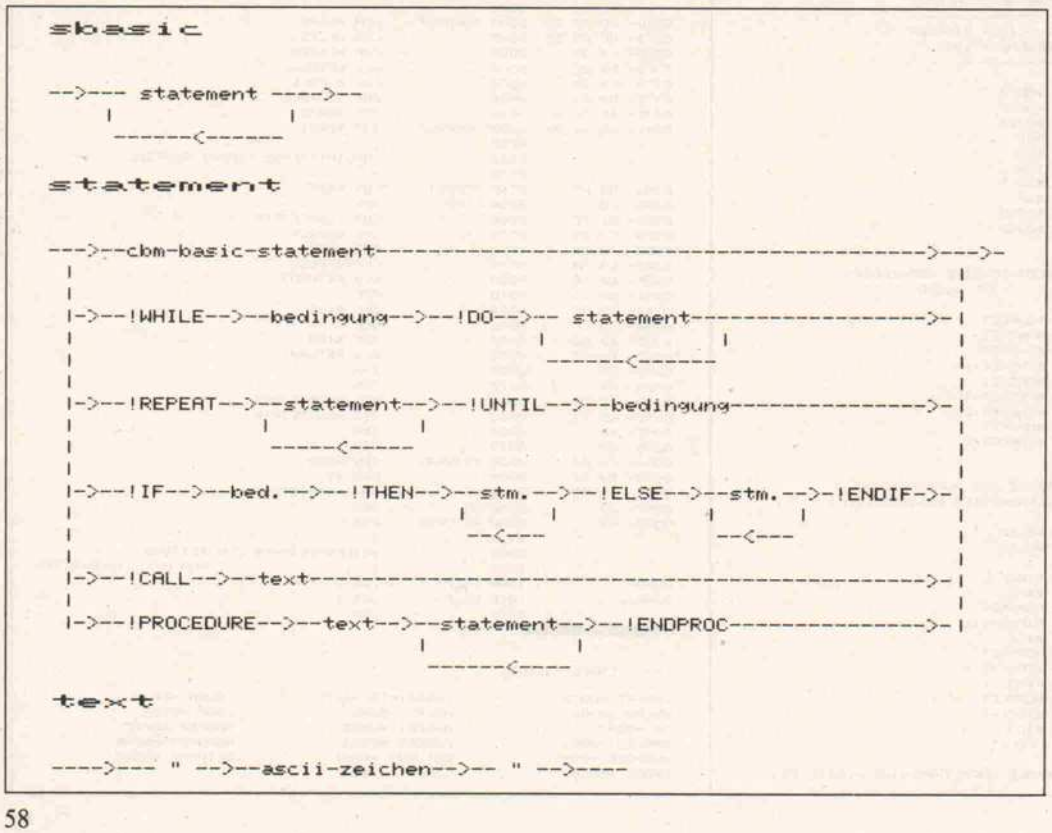
1) Beim übersetzten Programm im Zielfile sind die Linkpointer der BASIC-Zeilen noch nicht gesetzt. Dies ist normalerweise nicht schlimm, da die Pointer beim Laden von der Diskette automatisch gesetzt werden. Falls es aus irgendeinem Grund (z. B. beim Compilieren) aber doch stören sollte, so sind im Direktmodus folgende Befehle auszuführen:

```
LOAD 'Lw:Name',8
(Lw = Laufwerk)
CLR
SAVE 'QLw:Name',8
```

Diese Befehlsfolge besorgt das 'Relinken' des Programms.

2) Bei der Erstellung von SBASIC stand nicht der Faktor Geschwindigkeit im Vordergrund, sondern es wurde Wert darauf gelegt, den Preprozessor möglichst übersichtlich zu programmieren. Eine Umstellung von häufig verwendeten Unterprogrammen an den Programmanfang würde eine wesentliche Geschwindigkeitserhöhung bewirken. Das Programm kann natürlich auch mittels Compilierung schneller gemacht werden.

Wie übersichtlich ein SBasic-Programm ist, zeigt ein Beispielprogramm, das die Möglichkeiten der neuen Sprachelemente verdeutlicht (Tabelle 2).



CRS$(128)	: END
CRS$(137)	: GOTO
CRS$(139)	: IF
CRS$(141)	: GOSUB
CRS$(142)	: RETURN
CRS$(143)	: REM
CRS$(167)	: THEN
CRS$(168)	: NOT

Tabelle 4. Befehls-'Tokens'

Verwendete Literatur:

(1) Reinhold Martin, Dieter Smode: ROM und RAM bei PET und CBM, Franzis-Sonderheft 33

(2) Adam Osborne, Carroll S. Donahue: CBM-Computerhandbuch, tewi-Verlag

(3) Rodnay Zaks: Programmierung des 6502, Sybex Verlag

(4) S. Wittig, G. Deges: Struktolister und Programmentzerrer, elrad Heft 12/82

(5) Holger Petersen: Strukta, ein Preprozessor für Assembler, c't Heft 12/83

```
100 REM  ****************************
110 REM  *    BASICLOADER FUER     *
120 REM  *         'SPACES'        *
130 REM  *    ------------------   *
140 REM  *   (C) OLIVER FISCHER    *
150 REM  *                         *
160 REM  *      DEZEMBER 1983      *
170 REM  ****************************
180 :
190 :
200 :
210 PRINT "■    BASICLOADER FUER 'SPACES'"
220 PRINT "    ------------------------"
230 FOR I=826 TO 936
240 READ A
250 S=S+A
260 POKE I,A
270 NEXT I
280 IF S<>13498 THEN PRINT "■■■FEHLER IN DEN DATAZEILEN !■■■■■■■■";END
290 PRINT "■■■'SPACES' KANN JETZT MIT ■SYS 826■"
300 PRINT "■GESTARTET WERDEN !■■■■■■"
310 END
320 DATA  165,121,73,133,133
330 DATA  121,165,122,73,119
340 DATA  133,122,165,123,73
350 DATA  179,133,123,96,141
360 DATA  169,3,142,170,3
370 DATA  186,189,1,1,201
380 DATA  169,208,21,189,2
390 DATA  1,201,200,208,14
400 DATA  189,3,1,201,173
410 DATA  208,7,189,4,1
420 DATA  201,195,240,20,173
430 DATA  169,3,174,170,3
440 DATA  201,58,176,46,201
450 DATA  32,208,3,76,112
460 DATA  0,76,129,0,160
470 DATA  255,200,177,119,201
480 DATA  32,240,249,201,58
490 DATA  176,16,56,233,48
500 DATA  56,233,208,176,8
510 DATA  170,152,101,119,133
520 DATA  119,138,96,192,0
530 DATA  240,2,230,119,56
540 DATA  96
```

Tabelle 3. BASIC-Loader für 'SPACES'

```
100 REM  ****************************
110 REM  *   PRECOMPILER  SBASIC    *
120 REM  *   --------------------   *
130 REM  *                         *
140 REM  *    (C) OLIVER FISCHER   *
150 REM  *                         *
160 REM  * VERSION 1.5    DEZEMBER  *
170 REM  *                  1983   *
180 REM  ****************************
190 :
200 :
210 :
220 DIM TK$(11),TK%(11),W1(30),W2(50)
230 DIM R1(30),I1(50),I2(50),P1$(50),P1(50)
240 :
250 REM  --------------------
260 REM  BEFEHLSFELD BELEGEN
270 REM  --------------------
280 :
290 TK$(0)="WHILE":TK$(1)="DO":TK$(2)=CHR$(128)+"WHILE"
300 TK$(3)="REPEAT":TK$(4)="UNTIL"
310 TK$(5)=CHR$(139):TK$(6)=CHR$(167)
320 TK$(7)="ELSE":TK$(8)=CHR$(128)+CHR$(139)
330 TK$(9)="CALL":TK$(10)="PROCEDURE":TK$(11)=CHR$(128)+"PROC"
340 :
350 DATA  5,2,6,6,5,1,1,4,2,4,9,5:REM  LAENGE DER BEFEHLSWOERTER
360 FOR I=0 TO 11
370 READ TK%(I)
380 NEXT I
390 :
400 TRUE=-1:FALSE=0
410 LP$=CHR$(1)+CHR$(1)
420 LE$=CHR$(0)
430 PE$=LE$+LE$+LE$
440 CR$="■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■":REM  23 MAL CURSOR RECHTS
450 CU$="■■■■■■■■■■":REM  10 MAL CURSOR NACH OBEN
460 :
470 REM  ----------------------
480 REM  BEFEHLSKANAL EROEFFNEN
490 REM  ----------------------
500 :
510 OPEN 15,8,15
520 :
530 PRINT "■";TAB(10);"■SBASIC■ - PRECOMPILER"
540 PRINT "■";TAB(6);"COPYRIGHT BY  OLIVER FISCHER"
550 PRINT "■=========================================="
560 INPUT "■QUELLDATEI        ■■■";QD$
570 IF QD$="_" THEN PRINT "■■";:GOTO 560
580 INPUT "■AUF DISK 0/1    0■■■";D0$
590 IF D0$<>"0" AND D0$<>"1" THEN PRINT "■■";:GOTO 580
600 INPUT "■ZIELDATEI         ■■■";ZD$
610 IF ZD$="_" THEN PRINT "■■";:GOTO 600
620 INPUT "■AUF DISK 0/1    0■■■";DZ$
630 IF DZ$<>"0" AND DZ$<>"1" THEN PRINT "■■";:GOTO 620
640 INPUT "■EINGABEN ■OK■      ■■■";A$
650 IF LEFT$(A$,1)<>"J" AND LEFT$(A$,1)<>"Y" THEN PRINT CU$;:GOTO 560
660 PRINT "■=========================================="
670 :
680 REM  -------------------------
690 REM  LAUFWERKE INITIALISIEREN
700 REM  -------------------------
710 :
720 IF D0$=DZ$ THEN PRINT# 15,"I":D0$:GOTO 790
730 PRINT# 15,"I"
740 :
750 REM  ------
760 REM  PASS 1
770 REM  ------
780 :
790 GOSUB,1680
800 OPEN 2,8,2,CHR$(34)+D0$+":"+QD$+",P,R"+CHR$(34)
810 GOSUB 1680
820 OPEN 3,8,3,CHR$(34)+DZ$+":"+ZD$+"#,P,W"+CHR$(34)
830 GOSUB 1680
840 PRINT "■■PASS 1/ZEILE:■"
850 GOSUB 2080
860 GOSUB 1790
870 LP=A
880 IF LP=0 THEN PRINT# 3,PE$:GOTO 1180
890 GOSUB 1790
900 NZ=A:NH=A2:NL=A1
910 ZL$="":ZN=NZ:H=NH:L=NL
920 ZN$=CHR$(L)+CHR$(H)
930 AF=FALSE
940 PRINT "■";LEFT$(CR$,15);ZN
950 GOSUB 1980
960 IF A$=CHR$(34) OR A$=CHR$(143) THEN AF=NOT (AF)
970 IF AF THEN 990
980 IF A$=" " THEN 950
990 ZL$=ZL$+A$
1000 IF A$<>LE$ THEN 950
1010 GOSUB 1790
1020 LP=A
1030 IF LP=0 THEN EN=TRUE:GOTO 1060
1040 GOSUB 1790
1050 NZ=A:NH=A2:NL=A1
1060 IF LEFT$(ZL$,1)<>"!" THEN 1130
1070 I=0
1080 IF MID$(ZL$,2,TK%(I))=TK$(I) THEN 1110
1090 IF I<11 THEN I=I+1:GOTO 1080
1100 GOTO 1130
1110 ON (I+1)GOTO 2580,2700,2780,2920,3010,3130
1120 ON (I-5)GOTO 3260,3340,3460,3570,3690,3830
1130 PRINT# 3,LP$;ZN$;
1140 PRINT# 3,ZL$;
1150 ERR=FALSE
1160 IF NOT (EN) THEN 910
1170 PRINT# 3,PE$
1180 PRINT "■";CR$;RIGHT$("  "+STR$(ER%),3);" ERROR(S)    "
1190 CLOSE 2:CLOSE 3
1200 :
1210 REM  ------
1220 REM  PASS 2
1230 REM  ------
1240 :
1250 EN=FALSE
1260 ERR=FALSE
1270 ER%=0
1280 W3%=W2%-1:I3%=I1%-1:I4%=I2%-1
1290 W2%=0:I1%=0:I2%=0
1300 OPEN 2,8,2,CHR$(34)+DZ$+":"+ZD$+"#,P,R"+CHR$(34)
1310 GOSUB 1680
1320 OPEN 3,8,3,CHR$(34)+DZ$+":"+ZD$+",P,W"+CHR$(34)
1330 GOSUB 1680
1340 PRINT "■■PASS 2/ZEILE:■"
1350 GOSUB 2080
1360 GOSUB 1790
1370 LP=A
1380 IF LP=0 THEN PRINT# 3,PE$:GOTO 1590
1390 GOSUB 1790
1400 NZ=A:NH=A2:NL=A1
1410 ZL$="":ZN=NZ:H=NH:L=NL
1420 ZN$=CHR$(L)+CHR$(H)
1430 PRINT "■";LEFT$(CR$,15);ZN
1440 GOSUB 1980
1450 ZL$=ZL$+A$
1460 IF A$<>LE$ THEN 1440
1470 GOSUB 1790
1480 LP=A
1490 IF LP=0 THEN EN=TRUE:GOTO 1520
1500 GOSUB 1790
1510 NZ=A:NH=A2:NL=A1
1520 GOSUB 2400
1530 ON FLGOTO 3900,4010,4120,4230
1540 PRINT# 3,LP$;ZN$;
1550 PRINT# 3,ZL$;
1560 IF NOT (EN) THEN 1410
1570 PRINT# 3,PE$
1580 PRINT "■";CR$;RIGHT$("  "+STR$(ER%),3);" ERROR(S)    "
1590 CLOSE 2:CLOSE 3
1600 PRINT# 15,"S";DZ$;":";ZD$;"#"
1610 CLOSE 15
1620 END
1630 :
1640 REM  ---------------------
1650 REM  UP. DISK-ERROR-TEST
1660 REM  ---------------------
1670 :
1680 INPUT# 15,F1,F2$,F3,F4
1690 IF F1=0 THEN RETURN
1700 PRINT "■■■■■■■■■■■■■";LEFT$(CR$,20);
1710 PRINT "■";F2$;"■";"■■■■■■■■"
1720 CLOSE 2:CLOSE 3:CLOSE 15
1730 END
1740 :
1750 REM  ------------------------------
1760 REM  UP. 2-BYTE-ADRESSE EINLESEN
1770 REM  ------------------------------
1780 :
1790 GOSUB 1890
1800 A2=ASC(A$)
1810 A1=ASC(A1$)
1820 A=256*A2+A1
1830 RETURN
1840 :
1850 REM  ------------------------------
1860 REM  UP. 2-ZEICHEN-STRING EINLESEN
1870 REM  ------------------------------
1880 :
1890 GOSUB 1980
1900 A1$=A$
1910 GOSUB 1980
1920 RETURN
1930 :
1940 REM  ------------------------------
1950 REM  UP. ZEICHEN VON DISK EINLESEN
1960 REM  ------------------------------
1970 :
1980 GET #2,A$
1990 GOSUB 1680
2000 IF A$="" THEN A$=LE$
2010 RETURN
2020 :
2030 :
2040 REM  ------------------------------
2050 REM  UP.PROGRAMMHEADER UEBERTRAGEN
2060 REM  ------------------------------
2070 :
2080 GOSUB 1890
```

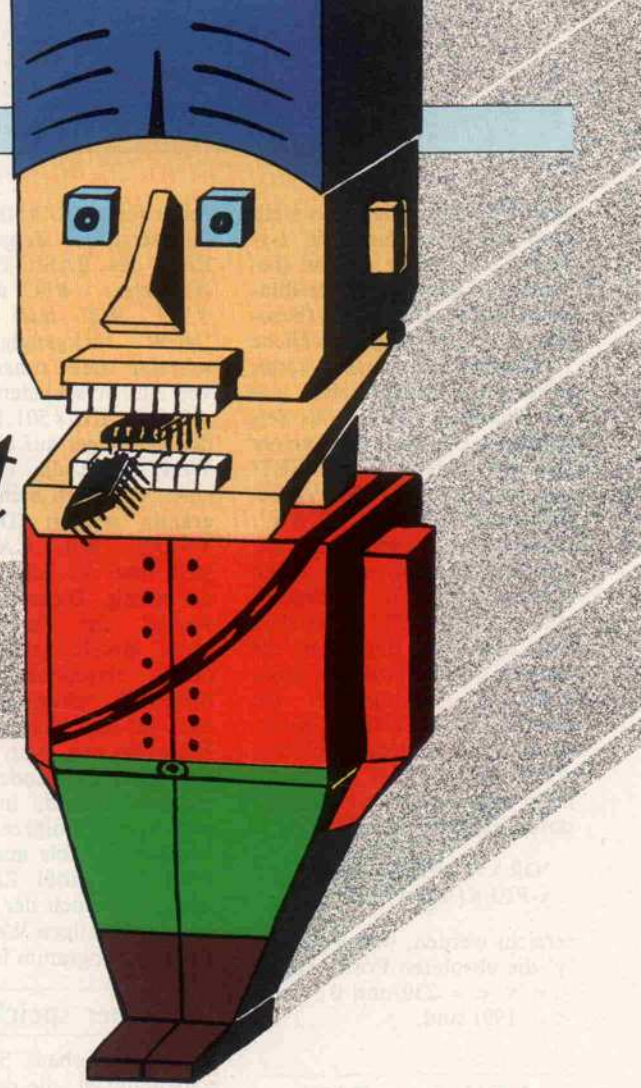
```
2090 PRINT# 3,A1$;A$;
2100 RETURN
2110 :
2120 REM  ----------------------
2130 REM  UP. TEILSTRING SUCHEN
2140 REM  ----------------------
2150 :
2160 :
2170 J=ANF
2180 L2=LEN(SC$)
2190 L1=LEN(ZL$)-1
2200 PO=0
2210 IF J>(L1-L2+1) THEN ERR=TRUE:GOTO 2240
2220 IF MID$(ZL$,J,L2)<>SC$ THEN J=J+1:GOTO 2210
2230 PO=J
2240 RETURN
2250 :
2260 :
2270 REM  ------------------------------
2280 REM  UP. ZAHL IN STRING VERWANDELN
2290 REM  ------------------------------
2300 :
2310 S1$=STR$(ZA)
2320 S1=LEN(S1$)-1
2330 ST$=RIGHT$(S1$,S1)
2340 RETURN
2350 :
2360 REM  ------------------------------
2370 REM  UP. FEHL. ZEILENNR. ERGAENZEN
2380 REM  ------------------------------
2390 :
2400 J=1:FL=0:AF=FALSE
2410 ZL=LEN(ZL$)
2420 IF J>ZL-1 THEN 2520
2430 Z$=MID$(ZL$,J,2)
2440 IF LEFT$(Z$,1)=CHR$(34) THEN AF=NOT (AF)
2450 IF AF THEN 2500
2460 IF Z$=CHR$(167)+"@" THEN FL=1:GOTO 2520
2470 IF Z$=CHR$(167)+"@" THEN FL=2:GOTO 2520
2480 IF Z$=CHR$(137)+"@" THEN FL=3:GOTO 2520
2490 IF Z$=CHR$(141)+CHR$(34) THEN FL=4:GOTO 2520
2500 J=J+1
2510 GOTO 2420
2520 PO=J:RETURN
2530 :
2540 REM  -----
2550 REM  WHILE
2560 REM  -----
2570 :
2580 SC$="!DO":ANF=7:GOSUB 2170
2590 IF ERR THEN ER%=ER%+1:PRINT "";CR$;"WHILE-DO   ERROR":GOTO 1130
2600 W1(W1%)=ZN
2610 W1%=W1%+1:W2%=W2%+1
2620 PRINT# 3,LP$;ZN$;CHR$(139);CHR$(168);"(";
2630 PRINT# 3,MID$(ZL$,7,PO-7);")";CHR$(167);"@";LE$;
2640 GOTO 1150
2650 :
2660 REM  --
2670 REM  DO
2680 REM  --
2690 :
2700 PRINT "";CR$;"WHILE-DO   ERROR"
2710 ER%=ER%+1
2720 GOTO 1130
2730 :
2740 REM  --------
2750 REM  ENDWHILE
2760 REM  --------
2770 :
2780 W1%=W1%-1
2790 PRINT# 3,LP$;ZN$;CHR$(137);
2800 ZA=W1(W1%):GOSUB 2310
2810 PRINT# 3,ST$;LE$;
2820 J=W2%-1
2830 IF J<0 THEN ER%=ER%+1:PRINT "";CR$;"ENDWHILE   ERROR":GOTO 1130
2840 IF W2(J)<>0 THEN J=J-1:GOTO 2830
2850 W2(J)=NZ
2860 GOTO 1150
2870 :
2880 REM  ------
2890 REM  REPEAT
2900 REM  ------
2910 :
2920 R1(R1%)=NZ
2930 R1%=R1%+1
2940 PRINT# 3,LP$;ZN$;CHR$(143);LE$;
2950 GOTO 1150
2960 :
2970 REM  -----
2980 REM  UNTIL
2990 REM  -----
3000 :
3010 R1%=R1%-1
3020 IF R1%<0 THEN ER%=ER%+1:PRINT "";CR$;"REP.-UNTIL ERROR":GOTO 1130
3030 PRINT# 3,LP$;ZN$;CHR$(139);CHR$(168);"(";
3040 PRINT# 3,MID$(ZL$,7,LEN(ZL$)-7);")";CHR$(167);
3050 ZA=R1(R1%):GOSUB 2310
3060 PRINT# 3,ST$;LE$;
3070 GOTO 1150
3080 :
3090 REM  --
3100 REM  IF
3110 REM  --
3120 :
3130 SC$="!"+CHR$(167)
3140 ANF=4
3150 GOSUB 2170
3160 IF ERR THEN ER%=ER%+1:PRINT "";CR$;"IF-THEN    ERROR":GOTO 1130
3170 I1%=I1%+1
3180 PRINT# 3,LP$;CHR$(L);CHR$(H);CHR$(139);CHR$(168);"(";
3190 PRINT# 3,MID$(ZL$,3,PO-3);")";CHR$(167);"@";LE$;
3200 GOTO 1150
3210 :
3220 REM  ----
3230 REM  THEN
3240 REM  ----
3250 :
3260 PRINT "";CR$;"IF-THEN    ERROR"
3270 ER%=ER%+1
3280 GOTO 1130
3290 :
3300 REM  ----
3310 REM  ELSE
3320 REM  ----
3330 :
3340 PRINT# 3,LP$;ZN$;CHR$(137);"@";LE$;
3350 J=I1%-1
3360 IF J<0 THEN ER%=ER%+1:PRINT "";CR$;"IF/ELSE    ERROR":GOTO 1130
3370 IF I1(J)<>0 THEN J=J-1:GOTO 3360
3380 I1(J)=NZ
3390 I2%=I2%+1
3400 GOTO 1150
3410 :
3420 REM  -----
3430 REM  ENDIF
3440 REM  -----
3450 :
3460 J=I2%-1
3470 IF J<0 THEN ER%=ER%+1:PRINT "";CR$;"IF/ENDIF   ERROR":GOTO 1130
3480 IF I2(J)<>0 THEN J=J-1:GOTO 3470
3490 I2(J)=NZ
3500 PRINT# 3,LP$;ZN$;CHR$(143);LE$;
3510 GOTO 1150
3520 :
3530 REM  ----
3540 REM  CALL
3550 REM  ----
3560 :
3570 SC$=CHR$(34):ANF=6:GOSUB 2170
3580 IF ERR THEN ER%=ER%+1:PRINT "";CR$;"CALL PROCD.ERROR":GOTO 1130
3590 HILF=PO
3600 ANF=PO+1:GOSUB 2170
3610 PRINT# 3,LP$;ZN$;CHR$(141);
3620 PRINT# 3,CHR$(34);MID$(ZL$,HILF+1,PO-HILF-1);LE$;
3630 GOTO 1150
3640 :
3650 REM  ---------
3660 REM  PROCEDURE
3670 REM  ---------
3680 :
3690 SC$=CHR$(34):ANF=11:GOSUB 2170
3700 IF ERR THEN ER%=ER%+1:PRINT "";CR$;"PROCEDURE  ERROR":GOTO 1130
3710 HILF=PO
3720 ANF=PO+1:GOSUB 2170
3730 P1$(P1%)=MID$(ZL$,HILF+1,PO-HILF-1)
3740 P1(P1%)=NZ
3750 P1%=P1%+1
3760 PRINT# 3,LP$;ZN$;CHR$(143);LE$;
3770 GOTO 1150
3780 :
3790 REM  -------
3800 REM  ENDPROC
3810 REM  -------
3820 :
3830 PRINT# 3,LP$;ZN$;CHR$(142);LE$;
3840 GOTO 1150
3850 :
3860 REM  -------------------
3870 REM  WHILE/UNTIL PASS 2
3880 REM  -------------------
3890 :
3900 PRINT# 3,LP$;ZN$;
3910 IF W2%>W3% THEN ER%=ER%+1:PRINT "";CR$;"WHILE-DO   ERROR":GOTO 1550
3920 ZA=W2(W2%):GOSUB 2310
3930 PRINT# 3,LEFT$(ZL$,PO);ST$;LE$;
3940 W2%=W2%+1
3950 GOTO 1560
3960 :
3970 REM  --------------
3980 REM  IF/THEN PASS 2
3990 REM  --------------
4000 :
4010 PRINT# 3,LP$;ZN$;
4020 IF I1%>I3% THEN ER%=ER%+1:PRINT "";CR$;"IF-THEN    ERROR":GOTO 1550
4030 ZA=I1(I1%):GOSUB 2310
4040 PRINT# 3,LEFT$(ZL$,PO);ST$;LE$;
4050 I1%=I1%+1
4060 GOTO 1560
4070 :
4080 REM  --------------
4090 REM  IF/ELSE PASS 2
4100 REM  --------------
4110 :
4120 PRINT# 3,LP$;ZN$;
4130 IF I2%>I4% THEN ER%=ER%+1:PRINT "";CR$;"IF/ELSE    ERROR":GOTO 1550
4140 ZA=I2(I2%):GOSUB 2310
4150 PRINT# 3,LEFT$(ZL$,PO);ST$;LE$;
4160 I2%=I2%+1
4170 GOTO 1560
4180 :
4190 REM  ------------------------------
4200 REM  CALL/PROCEDURE/ENDPROC PASS 2
4210 REM  ------------------------------
4220 :
4230 J=0
4240 P1%=P1%-1:HILF$=MID$(ZL$,PO+2,LEN(ZL$)-PO-2)
4250 IF J>P1% THEN ER%=ER%+1:PRINT "";CR$;"CALL PROCD.ERROR":GOTO 1540
4260 IF P1$(J)<>HILF$ THEN J=J+1:GOTO 4250
4270 PRINT# 3,LP$;ZN$;CHR$(141);
4280 ZA=P1(J):GOSUB 2310
4290 PRINT# 3,ST$;LE$;
4300 GOTO 1560
READY.
```

Wichtige Variablen von 'SBASIC'

```
TK$( )          : Neue Befehlswoerter
TK%( )          : Laenge der Befehlswoerter
W1( ),W2( )     : Stacks fuer WHILE - ENDWHILE
R1( )           : Stack fuer REPEAT - UNTIL
I1( ),I2( )     : Stacks fuer IF - THEN - ELSE - ENDIF
P1$( )          : Namen der Prozeduren im Quellprogramm
P1( )           : Zeilennummern der Prozedurbeginne
LP$             : Pseudo - Linkpointer
LE$             : Zeichen fuer Zeilenende
PE$             : Zeichenfolge fuer Programmende
QD$             : Name der Quelldatei
DQ$             : Nummer der Quelldisk
ZD$             : Name der Zieldatei
DZ$             : Nummer der Zieldisk
LP              : Linkpointer
ZN,ZN$          : Nummer der gerade bearbeiteten Zeile
NZ              : Naechste Zeilennummer
ZL$             : Gerade bearbeitete BASIC - Zeile
AF              : Flag fuer Anfuehrungszeichen
ERR             : Flag fuer Fehler
EN              : Flag fuer Programmende
ER%             : Fehlerzahl
F1,F2$,F3,F4    : Variablen fuer Diskfehler
FL              : Flag fuer 'Zeilennummer ergaenzen' (Pass2)
```

ORIC-ROM geknackt

Teil 1



Das BASIC des ORIC-1 bietet viele Möglichkeiten, die man bei anderen Computern dieser Preisklasse nicht findet. Allerdings ist es auch den Vätern des ORIC nicht gelungen, alle möglichen Forderungen an das BASIC zu befriedigen. Mit diesem Artikel verraten wir einige nützliche Tricks. Wer seinen ORIC bis ins ROM durchschauen möchte, findet in diesem und im nächsten Heft Listen aller wichtigen Adressen vor.

Ekkehard Otto

Der Bildschirm des ORIC ist zu 28 Zeilen mit je 40 Zeichen organisiert. Allerdings kann normalerweise eine Zeile nicht die 40 Zeichen aufnehmen, da die ersten beiden Zeichen einer Zeile die Informationen für die Vordergrund- und die Hintergrundfarbe enthalten. Außerdem kann die erste Zeile nicht beschriftet werden, weil sie als Statuszeile reserviert ist.

Bildschirm im TEXT-MODE

Diese beiden Einschränkungen lassen sich jedoch umgehen. Schaltet man mit den Befehlen 'PAPER 0 : INK7' das Bild auf 'weiße Schrift auf schwarzem Grund' und gibt mit 'CTRL]' die ersten beiden Spalten frei, können alle 40 Zeichen einer Zeile benutzt werden. Allerdings sollte man dann die Befehle 'INK' und 'PAPER' nicht mehr anwenden, da dies die Daten in den vorderen Spalten einer Zeile löscht.

Will man Informationen in der Statuszeile darstellen, so kann man dies mit einer Folge von POKE-Befehlen erreichen. Die Statuszeile fängt bei der Speicheradresse 48000 an, so daß:

```
FOR I=1 TO LEN(A$)
:POKE48000+I,MID$
(A$,I,1):NEXT I
```

den STRING A$ in der Statuszeile anzeigt. Einfacher geht es jedoch mit der Befehlsfolge:

```
DOKE#1F,48000:
POKE#2F8,0:
CALL#D9C9,A$
```

Außerdem ist die vorgegebene Einteilung in eine feste Statuszeile und 27 Textzeilen auch veränderbar. Dazu dienen die Speicherplätze #26D—#26F. In #26D/#26E steht die Anfangsadresse des Bildschirmbereichs minus 40 (normalerweise 48000), und in #26F steht die Anzahl der Bildschirmzeilen, die für 'PRINT', 'PLOT' sowie 'LIST' zur Verfügung stehen. Wird der Wert in #26D/#26E um 40 erhöht (DOKE#26D, 48040), so wird eine weitere Zeile wie die Statuszeile geschützt.

Allerdings würde das jetzt bedeuten, daß der Bildschirm insgesamt 29 Zeilen umfassen würde. (2 Statuszeilen + 27 [#26F] sonstige Zeilen.) Dafür gibt es aber nicht genug Bildschirmspeicher, so daß beim 'Scrollen' (Hochschieben des Bildschirminhaltes, wenn die unterste Zeile beschrieben wurde) merkwürdige Zeichen auf dem Bildschirm erscheinen. Deshalb muß der Wert in #26F entsprechend geändert werden (POKE#26F,26). Die Anzahl der am Anfang geschützten Zeilen plus der Wert in #26F darf 28 nicht überschreiten. Verkleinert man nun den Wert in #26F weiter, so werden auch noch am unteren Bildrand Zeilen geschützt. Mit:

```
DOKE#26D,48120:
POKE#26F,22
```

werden zum Beispiel die oberen vier und die beiden letzten Zeilen geschützt. Die so geschützten Zeilen werden durch 'CLS' und 'CTRL L' nicht gelöscht und können nur durch 'PLOT' oder 'POKE' beschrieben werden.

Damit verfügt man nun über ein Bildschirmfenster, in dem feste Informationen in den oberen und unteren Zeilen stehen, während der Rest des Bildschirms wie üblich 'scrollt'.

Bei manchen Computern gibt es die Möglichkeit, den Cursor an eine bestimmte Stelle des Bildschirms zu setzen (PRINT AT). Beim ORIC-1 kann man zwar mit Hilfe von 'PLOT' die Ausgabe an beliebiger Stelle steuern, der Cursor bleibt aber wo er ist. Das ist zum Beispiel dann unerwünscht, wenn man Eingaben in einer Bildschirmmaske vornehmen will.

Nun steht zwar die augenblickliche Cursorposition in den Speicherstellen #268 (Spalte 0 < = s < = 39) und #269 (Zeile 1 < = z < = 27); trotzdem ist nicht möglich, durch Veränderung dieser Werte die Position des Cursors zu bestimmen, da noch andere Werte verändert werden müssen. Dies kann durch Verwendung von Systemroutinen geschehen. Dabei muß dafür gesorgt werden, daß der Cursor nicht als schwarzer Fleck am alten Platz stehenbleibt. (Auch das vom Handbuch vorgeschlagene Maschinenprogramm hat diesen Fehler.) Will man in Spalte 's' und Zeile 'z' schreiben, so kann man dazu folgende Befehle verwenden.

```
'DOKE#261,#F6FF:
CALL#F5E6:POKE#269,s:
POKE#268,z:
CALL#F67D:PRINT .....'
```

Die ersten beiden Befehle schalten dabei den Cursor und den Interrupt ab. Deshalb müssen im Direktmodus alle Befehle in einer Zeile stehen. In einem Programm darf nach den ersten beiden Befehlen nur kein 'INPUT', 'GET' oder 'WAIT' stehen. Ist der Cursor im Programm abgeschaltet, so kann man den Teil bis 'CALL#F5E6' weglassen, da der erste Teil nur verhindern soll, daß der Cursor als schwarzer Fleck an der alten Stelle stehenbleibt.

Folgender 'Befehl' ist zwar im ORIC-1 vorgesehen, wird aber im Handbuch nicht erwähnt: 'PRINT CHR$(30)' setzt den CURSOR in die linke obere Ecke des Bildschirms, ohne den Schirminhalt zu löschen (HOME).

... und im HIRES-Mode

Wenn man sich die Graphik-Befehle (CURSET, CURMOV

und DRAW) ansieht, so stellt man fest, daß man die x-y-Position entweder absolut (bezogen auf die linke obere Bildschirmecke) oder relativ (bezogen auf die augenblickliche Cursorposition) angeben kann. Diese zwei Möglichkeiten gelten allerdings nur für das Setzen (bzw. Löschen) des Cursors mit den Befehlen 'CURSET' (absolut) und 'CURMOV' (relativ), nicht aber für 'DRAW'. Dies ist zum Beispiel dann unangenehm, wenn man einen Funktionsgraphen durchgezogen mit 'DRAW' darstellen will. Man muß dann stets die aktuelle Cursorposition (inkl. Rundung) 'nachhalten'. Da sich jedoch die 'HIRES'-Koordinaten in den Speicherplätzen #219 und #21A befinden, kann das absolute 'DRAW' durch die Befehlsfolge

'DRAWx-PEEK(#219),
y-PEEK(#21A),FB'

erreicht werden, wobei 'x' und 'y' die absoluten Positionen (0 <= x <= 239 und 0 <= y <= 199) sind.

NEW statt RUN

Wer schon einmal aus Versehen 'NEW' statt 'RUN' getippt hat (nach Eintippen eines langen Programms), wird den folgenden kleinen Trick begrüßen.

Ein irrtümliches 'NEW' kann nämlich durch die folgenden Befehle wieder rückgängig gemacht werden:

```
POKE#502,1:DOKE#9C,
#9700:CALL#C56F:CLEAR
I=#501:REPEAT:I=
DEEK(I):UNTILDEEK(I)=
0:DOKE#9C,I+2
```

Will man verstehen, wie diese Zeilen arbeiten, so muß man zunächst einmal feststellen, wie BASIC-Programme im Speicher stehen und was der Befehl 'NEW' bewirkt.

Ein Zeile eines BASIC-Programms beginnt im Speicher stets mit einem Zwei-Byte-Zeiger auf den Beginn der nächsten Zeile (Folgezeiger). Dann folgt die Zeilennummer (auch zwei Byte) und die eigentliche Zeile, die mit 'Null' endet. Am Programmende steht als Folgezeiger der Wert 'Null'.

Der Befehl 'NEW' löscht nun nicht etwa das Programm, sondern setzt nur den ersten Folgezeiger (in #501/#502) auf Null und setzt den Zeiger auf das Ende des BASIC-Programms (Endzeiger #9C/#9D) auf #503. Will man nun das 'NEW' rückgängig machen, schreibt man zunächst einen von Null verschiedenen Wert in #501 (POKE #501,1) und setzt den Endzeiger auf einen hohen Wert, damit die Verwendung einer Variablen nicht das Programm zerstört (DOKE#9C, #9700). Mit CALL#C56F stellt man die Folgezeiger wieder richtig. Diese Routine verwendet das Betriebssystem, wenn BASIC-Zeilen gelöscht oder eingegeben werden. 'CLEAR' schließlich bereitet alles für die Verwendung von Variablen vor. Nun muß man nur noch das Ende des Programms suchen, indem man sich an den Folgezeigern 'entlanghangelt', bis man auf den Wert 'Null' stößt. Zum Schluß wird dann noch der Endzeiger auf den richtigen Wert gesetzt, und das Programm ist gerettet.

Sicher speichern

Mit der hohen Speichergeschwindigkeit, die der ORIC-1 im 'FAST-Modus' für Kassettenbetrieb bietet, verfügt der ORIC-Benutzer über einen schnellen externen Speicher. Es gibt Floppy-Disks, die kaum schneller sind. Wenn man einen guten Rekorder benutzt, ist das Verfahren auch sicher. Trotzdem vermißt man die Möglichkeit, die Aufzeichnung und den Ladevorgang kontrollieren zu können. Diese Möglichkeit bietet ein Programm, das den Befehlssatz des ORIC-1 um drei Befehle erweitert.

!V überprüft, ob ein Programm richtig abgespeichert wurde.
Syntax : !V "Name"
oder : !V""

Für Programme, die im SLOW-Modus abgespeichert wurden, wird jeweils ',S' angehängt.

Meldet sich das Programm mit 'Ready' zurück, so stimmt die Bandaufzeichnung mit dem Speicherinhalt überein. Im anderen Fall meldet sich das Programm mit der Adresse der ersten Abweichung und der Meldung 'Load-error' zurück.

!C errechnet eine Prüfsumme für ein Basicprogramm.

!L errechnet die Prüfsumme eines Programms, das geladen oder mit !V überprüft wurde. Stimmt diese mit der vorher errechneten überein, so wurde fehlerfrei geladen.

Benutzung des Programms:
Nach dem Einschalten des ORIC wird das Programm geladen und mit

DOKE#2F5,1174'

eingeschaltet.

Soll nun ein neugeschriebenes Programm auf das Band gerettet werden, so geschieht das zunächst mit 'CSAVE'. Dann wird das Band zurückgespult und mit '!V"" überprüft. Die Kontrollsumme (!L) wird notiert. Bei einem späteren Laden kann dann durch Vergleich mit der Kontrollsumme der Ladevorgang überprüft werden.

Speicherbelegung

Viele nützlichen Routinen sind bereits im ROM des ORIC enthalten. Werden diese Routinen von eigenen Programmen aufgerufen, kann man viel Arbeit (und Speicherplatz) sparen. Um die ROM-Routinen anzusprechen, muß man aber wissen, welche Routine wo im ROM steht und was sie bewirkt. Das 'Wo' und 'Was' finden Sie in der nächsten c't.

```
#0400  20 25 E7  JSR  E725  ; *** VERIFY *** Parameter
#0403  20 CA E6  JSR  E6CA  ; Initialisation VIA 6522 für TAPE
#0406  20 63 E5  JSR  E563  ; Lösche Statuszeile
#0409  A9 03     LDA  #3    ; 'Searching'
#040B  A0 E5     LDY  #E5   ;
#040D  20 76 E5  JSR  E576  ;
#0410  20 96 E6  JSR  E696  ; Lies Vorspann zur Synchronisation
#0413  20 30 E6  JSR  E630  ; Lies ein Byte
#0416  C9 24     CMP  #24   ; = hex 24 ?
#0418  D0 F9     BNE  413   ; falls nein, zurück
#041A  A2 09     LDX  #9    ; Einlesen der Adressen
#041C  20 30 E6  JSR  E630  ; und sonstiger Informationen
#041F  95 5D     STA  5D,X  ; (AUTO etc.)
#0421  CA        DEX        ; vom Band
#0422  D0 F8     BNE  41C   ;
#0424  20 30 E6  JSR  E630  ; Einlesen des Namens
#0427  F0 05     BEQ  42E   ;
#0429  95 49     STA  49,X  ;
#042B  E8        INX        ;
#042C  D0 F6     BNE  424   ;
#042E  95 49     STA  49,X  ;
#0430  A2 16     LDX  #16   ; Anzeige des Namens
#0432  A9 49     LDA  #49   ; in der Statuszeile
#0434  20 73 E5  JSR  E573  ;
#0437  A5 5F     LDA  5F    ; Anfangsadresse
#0439  A4 60     LDY  60    ;
#043B  85 33     STA  33    ; nach 33/34
#043D  84 34     STY  34    ;
#043F  A0 00     LDY  #0    ;
#0441  20 30 E6  JSR  E630  ; Lies Byte
#0444  B0 0C     BCS  452   ; Fehler, falls ja —> Ende
#0446  D1 33     CMP  (33),Y ; vergleiche mit Speicher
#0448  D0 08     BNE  452   ; falls ungleich —> Ende
#044A  20 54 E5  JSR  E554  ; erhöhe 33/34 und prüfe auf Ende
#044D  90 F2     BCC  441   ; falls noch nicht fertig, weiter
#044F  4C 04 E8  JMP  E804  ; Löschen Status, Rücksetzen VIA, RETURN
#0452  A6 33     LDX  33    ; ENDE mit Fehler
#0454  A5 34     LDA  34    ; Adresse des Fehlers
#0456  85 D1     STA  D1    ;
#0458  86 D2     STX  D2    ;
#045A  20 C5 E0  JSR  E0C5  ; Ausdrucken
#045D  4C 3E E5  JMP  E53E  ; 'LOAD ERROR ...'
#0460  A5 9A     LDA  9A    ; *** C ***
#0462  A6 9B     LDX  9B    ; Anfang BASIC
#0464  85 5F     STA  5F    ;
#0466  86 60     STX  60    ; nach Programmanfangsadresse
#0468  A5 9C     LDA  9C    ; Ende BASIC
#046A  A6 9D     LDX  9D    ;
#046C  85 61     STA  61    ; nach Programmendadresse
#046E  86 62     STX  62    ;
#0470  A9 00     LDA  #0    ; *** L ***
#0472  85 D1     STA  D1    ; Löschen Kontrollsumme
#0474  85 D2     STA  D2    ;
#0476  A5 5F     LDA  5F    ; Anfang
#0478  A4 60     LDY  60    ; nach
#047A  85 33     STA  33    ; 33/34
#047C  86 34     STX  34    ;
#047E  18        CLC        ; Vorbereiten Addition
#047F  A0 00     LDY  #0    ;
#0481  B1 33     LDA  (33),Y ; Programmbyte aus Speicher
#0483  65 D2     ADC  D2    ; zu Summe
#0485  85 D2     STA  D2    ; addieren (low)
#0487  A9 00     LDA  #0    ;
#0489  65 D1     ADC  D1    ; (high)
#048B  85 D1     STA  D1    ;
#048D  20 54 E5  JSR  E554  ; (wie bei 44A)
#0490  90 EF     BCC  481   ;
#0492  4C C5 E0  JMP  E0C5  ; Ausdrucken der Summe
#0496  48        PHA        ; *** PROGRAMMANFANG *** Sichere Kennbuchstaben
#0497  20 E2 00  JSR  E2    ; Zeichenzeiger auf nächstes Zeichen setzen.
#049A  68        PLA        ; Kennbuchstaben zurückholen
#049B  EA        NOP        ; frei für Erweiterung
#049C  EA        NOP        ;
#049D  EA        NOP        ;
#049E  EA        NOP        ;
#049F  EA        NOP        ;

#04A0  C9 56     CMP  #56   ; falls Kennbuchstabe = 'V'
#04A2  D0 03     BNE  4A7   ;
#04A4  4C 00 04  JMP  400   ; Sprung nach *** VERIFY ***
#04A7  C9 43     CMP  #43   ; falls = 'C'
#04A9  F0 B5     BEQ  460   ; Sprung nach *** C ***
#04AB  C9 4C     CMP  #4C   ; falls = 'L'
#04AD  F0 C1     BEQ  470   ; Sprung nach *** L ***
#04AF  4C A0 D2  JMP  D2A0  ; 'Syntax Error'
```

Tabelle 1. Das Programm 'VERIFY'

Tabelle 2

Systemadressen des ORIC-1

0000-000B	frei
000C-0011	diverse Zeiger
0012-0013	Adresse des Beginns der aktuellen Bildschirmzeile
0014-0015	Zeiger für SOUND
0017	Länge INPUT
0018-0019	Zeiger auf Keywordtable
001A-001C	JMP #CBED Print 'READY'
001D-0020	diverse Zeiger
0021-0023	USR JMP
0024	Suchcharacter
0025	Hochkommaflag
0026	Pointer für INPUT
0027	Speicher für 1. Zeichen Namen bei DIM
0028	Flag über Variablenart (#00 = numerisch #FF-String)
0029	Flag für Integer (#80 = Integer #00 = sonst.)
002A	Flag für DATA und Hochkomma bei LIST
002B	Flag für DIM und FN
002C	Flag für INPUT (#00) GET (#40) oder READ (#98)
002D	Flag für Vorzeichen bei TAN und Vergleichen
002E	Flag für Unterdrückung der Ausgabe
002F	Zwischenspeicher für #-Zahlen
0030	Cursorposition auf Drucker
0031	Spaltenzahl Drucker
0032	letzte TAB-Position auf Drucker
0033-0034	BASIC-Adresse für CALL, GOTO etc.
0035-0084	BASIC-INPUT-PUFFER
005F-0060	Tape-Startadresse
0061-0062	Tape-Endadresse
0063	Tape-Auto-Flag (#04 = Auto)
0064	Flag für gegebene Programmstartadresse
0065-0066	?
0067	Flag für 'S'
0085	Zeiger auf nächsten Descriptor
0086-0087	Zeiger auf zuletzt verwendeten String
0091-0092	Transfer-Pointer
0093-0095	diverse Zeiger
009A-009B	Zeiger auf Beginn des BASIC-Text (1,h)
009C-009D	Zeiger auf Beginn der Variablentafel (= Ende des BASIC-Pgm)
009E-009F	Zeiger auf Beginn der ARRAY-Tafel
00A0-00A1	Zeiger auf Ende der ARRAY-Tafel
00A2-00A3	Zeiger auf Ende des letzten String
00A4-00A5	Zeiger auf Beginn des letzten String
00A6-00A7	Zeiger auf höchste RAM-Adresse
00A8-00A9	Nummer der aktuellen BASIC-Zeile
00A9	= #FF, falls direkter Befehl
00AA-00AB	Zeilennummer für CONT
00AC-00AD	Zeiger auf nächsten BASIC-Befehl
00AE-00AF	Nummer der aktuellen DATA-Zeile
00B0-00B1	Zeiger auf nächstes DATA
00B2-00B3	INPUT-Zeiger
00B4-00B5	Name der aktuellen Variablen 1. und 2. Zeichen
00B6-00B7	Adresse der aktuellen Variablen
00B8-00B9	Zeiger auf aktuelle FOR/NEXT-Variable
00BA-00BB	Zeiger auf aktuellen Operator
00BC	Maske für Operator (< = >)
00BD-00C2	diverse Zwischenspeicher
00C3-00C5	JMP für Funktionen
00C6-00CA	Zwischenspeicher 1 für Funktionsargumente
00CA-00CF	Zwischenspeicher 2 für Funktionsargumente
00D0-00D5	Fließkommaakku #1
00D6	Zähler für Funktionsauswertung
00D7	Zähler für Normalisierung Akku #1
00D8-00DD	Fließkommaakku #2
00DD	Vorzeichenvergleich beider Akkus
00DF	Rundungsbyte Akku #1
00E0-00E1	Zeiger
00E2-00F3	CHRGET-Routine (holt neues Zeichen aus BASIC-Text)
00E8	CHRGOT (holt letztes Zeichen)
00E9-00EA	Zeiger auf aktuelles BASIC-Zeichen
00FA-00FE	Zufallszahl für RND
00FF	Vorzeichen für ASCII-String
0100-01FF	Stack
0100-010A	Puffer für Binär/ASCII-Umwandlung
0200-0207	diverse Zwischenspeicher
0208	gerade gedrückte Taste (Sondercode)
0209	#A2:CTRL, #A4:SHIFT rechts, #A7:SHIFT links
020A-020B	diverse Zwischenspeicher
020C	Flag für Groß-/Kleinschreibung (#FF = CAPS, #7F = Klein)
020D-0211	Zähler für Tastaturabfrage
0212	FB (für Graphikbefehle)
0213	PATTERN
0214-0218	diverse Speicher
0219	HIRES X
021A	HIRES Y
021B-021E	diverse Speicher
021F	= 1, falls HIRES
0220	16/48-K-Flag = 0,falls 48K
0221-0227	frei
0228-022A	JMP für IRQ
022B-022D	JMP für NMI
022E-022F	?
0230	RTI für Rücksprung aus IRQ-Routine
0231-0260	frei, hier kann eine Ergänzung zur IRQ-Routine geschrieben werden. (0230 wird überschrieben, und die Ergänzung muß mit RTI abschließen. Benötigte Register müssen auf dem Stack gesichert werden.)
0261-0262	JMP-Zeiger für Controllcodeausführung
0263-0265	diverse Speicher
0266-0267	frei
0268	Cursorposition Zeile
0269	Cursorposition Spalte
026A	Flags BIT 0 = 0 : Cursor aus BIT 1 = 0 : Video aus BIT 2 = 1 : CNTRL P gedrückt BIT 3 = 1 : Tastenton aus BIT 4 = 1 : ESC BIT 5 = 0 : 1. und 2. Spalte geschützt BIT 6 = 1 : Doppelt schreiben
026B	Hintergrundfarbe
026C	Vordergrundfarbe
026D-026E	Bildschirmfensterstartadresse
026F	Anzahl Zeilen Bildschirmfenster
0271	Cursorflag
0272-0277	interne Zähler (Cursor-Blinken, WAIT etc.)
0278-02BF	frei
02C0	GRAB/RELEASE-Flag BIT 0 = 1 : HIRES, BIT 1 = 1 RELEASE
02C1-02C2	Startadresse HIRES-Screen
02C3	Parameter für SOUND und GRAPHIK-Befehle
02DF	letzte gedrückte Taste (ASCII + #80)
02E0	Fehlerflag für Sound/Graphik
02F1	Ausgabeflag (#80 = Drucker, #00 = Bildschirm)
02F2	Flag für RTS nach LIST (#80 kein Warmstart nach LIST)
02F4	TRACE-Flag (#80 = TRACE)
02F5-02F6	Adresse für !
02FC-02FD	Adresse für &
0300-030F	Register des VIA
0310-03FF	Wiederholung der Adressen 0300-030F
0400-04FF	frei
0500	BASIC-Textbereich
9800-9BFF	Standardzeichensatz (HIRES)
9C00-9FFF	alternativer Zeichensatz (HIRES)
A000-BFDF	Hires-Bildschirm
B400-B7FF	Standardzeichensatz (TEXT)
B800-BC7F	alternativer Zeichensatz (TEXT)
BB80-BFDF	Text-Bildschirm
BF40-BFDF	HIRES-Text
C000-FFFF	ROM-Bereich

CP/M-Tip

Laden ohne Starten

Holger Petersen

Unter der Überschrift 'CP/M-Tip' sollen in unregelmäßiger Reihenfolge Beiträge erscheinen, die den CP/M-Anwendern einige wenig bekannte Eigenarten des Betriebssystems zeigen sollen. CP/M-Freaks sind eingeladen, ihre Erfahrungen (selbstverständlich gegen Honorar) zu veröffentlichen. Der erste Tip zeigt, wie Programme in den Speicher geladen werden können, ohne daß sie sich automatisch selbst starten.

In c't 12/83 haben wir schon einmal auf die Möglichkeit hingewiesen, ein sich im Speicher befindendes Programm nach einem Abbruch neu zu starten. Diese Methode ist aber nicht nur sinnvoll, wenn Programme auf andere Benutzerebenen kopiert werden sollen, sondern kann auch angewandt werden, wenn wenig Diskettenplatz oder nur ein Laufwerk für Kopiervorgänge zur Verfügung steht.

Einige Programme können nach ihrem Aufruf nicht ohne Probleme verlassen und wieder neu angesprungen werden. In diesen Fällen kann man mit einem Trick ein Programm von der Diskette in den Speicher laden und es nicht ausführen lassen. In der Kommando-Zeile wird dazu hinter den Programmnamen ein Leerzeichen (Space) und dann ein 'Control C' eingegeben. Wird nun die 'Return'-Taste gedrückt, lädt das CP/M den File von der Floppy in den Speicher und findet danach als weiteren Befehl das 'Control C'. Dieses Zeichen wird als Fehler interpretiert, worauf der CCP die Fehlermeldung '?' ausgibt und wieder in den Eingabemodus verzweigt.

Nun kann man die Diskette wechseln. Nach der erneuten Eingabe von 'Control C' besteht die Möglichkeit, das Programm mit dem Befehl 'SAVE nn FILENAME.COM' wieder auf Diskette zu sichern oder mit einem vorher erzeugten File 'GO' (SAVE Ø GO.COM) erneut zu starten.

Der neue ORIC

Andreas Burgwitz

Erwartete man nach dem ORIC-1 den ORIC-2, so wird schon der Name des Neuen überraschen: ORIC ATMOS. Dahinter verbirgt sich allerdings auch kein neuer Computer, sondern ein Rechner, der in vielen Details gegenüber seinem Vorgänger verbessert ist.

Der ORIC ATMOS zeigt, daß ein Computerhersteller in der Lage ist, die Schwächen eines Modells zu erkennen und zu beheben. Die am ORIC-1 störenden Fehler im Betriebssystem und einige 'Schönheitsfehler' sind bei dem ATMOS beseitigt. Geblieben ist das Grundkonzept: 6502A CPU, 16 oder 48 KByte RAM und ein 16 K BASIC-ROM; hochauflösende Grafik mit acht Farben sowie ein programmierbarer Soundgenerator.

Der erste Blick

Wer den ORIC-1 kennt, wird am ATMOS als erstes die neue Tastatur bemerken. Der ORIC-1 war schon mit Kunststofftasten in der Standard-QWERTY-Anordnung ausgerüstet. Diese Tastatur war gut zu bedienen, entsprach in ihrer Größe aber nicht der Norm. Beim ATMOS ist dieser Nachteil beseitigt. Die einzelnen Tasten haben nun die gewohnte Größe, was für geübte 'Vielschreiber' eine wichtige Erleichterung sein dürfte.

Alle Tasten sind mit Autorepeat ausgestattet. Zu den schon beim ORIC-1 vorhandenen Tasten 'Escape' und 'Control' kommt beim ORIC ATMOS noch eine 'Function'-Taste hinzu.

Nach wie vor können mit der Control-Taste wichtige Steuerfunktionen, wie Bildschirmlöschen oder 'Break' ausgeführt werden. Der bei jeder Tastenbetätigung ausgelöste kurze Piepton läßt sich ebenfalls durch eine Control-Funktion ein- oder ausschalten.

Wie schon der ORIC-1, besitzt der ATMOS eine RESET-Taste, die von der Unterseite des Gehäuses her erreichbar ist. Der Computer kann damit ohne Programmverlust in einen definierten Zustand gebracht werden. Das kann nötig werden, wenn zum Beispiel das Laden von der Kassette Schwierigkeiten bereitet.

Kontakte

Der ORIC ATMOS verfügt, ebenso wie sein Vorgänger, über folgende Anschlüsse: UHF- und RGB-Ausgang, Anschluß für Kassettenrekorder und eine Centronics-Druckerschnittstelle. An eine weitere Steckerleiste können Hardwareerweiterungen angeschlossen werden, da der komplette Systembus an dieser Leiste liegt. Sämtliche Anschlüsse sind bei dem ATMOS bezeichnet, was bei dem ORIC-1 nicht der Fall war.

Betriebssystem

Die wichtigste Veränderung am ORIC ist nicht auf den ersten Blick zu bemerken: Der ATMOS ist mit einem neuen, wesentlich verbesserten Betriebssystem ausgerüstet. Der Anwender kann immer noch über sehr leistungsstarke, bei anderen Rechnern selten zu findende BASIC-Befehle verfügen, muß aber nicht mehr auf einige wichtige Anweisungen verzichten oder Fehler in Kauf nehmen.

So kann beim ATMOS mit einem Befehl festgestellt werden, ob ein Programm richtig auf Kassette gespeichert wurde. Auch können nun Variable auf Band geschrieben werden. Der ORIC-1 bot diese Möglichkeit nicht. Beibehalten wurde das Aufzeichnungsverfahren: Die Programme können nach wie vor mit 300 Baud oder 2400 Baud auf Band gespeichert werden. Damit ist sichergestellt, daß Programme vom ORIC-1 auch vom ATMOS gelesen werden können.

Beseitigt wurde auch ein Fehler im BASIC. Konnte der ORIC-1 die Befehle PRINT TAB(1) bis PRINT TAB(13) nicht richtig ausführen, so macht dies beim ATMOS keine Schwierigkeiten.

Das Editieren eines Programms ist, wie bei dem ORIC-1, etwas umständlich. So wird mancher Anwender lieber eine Zeile neu eintippen, als sie korrigieren.

Alles auf einmal

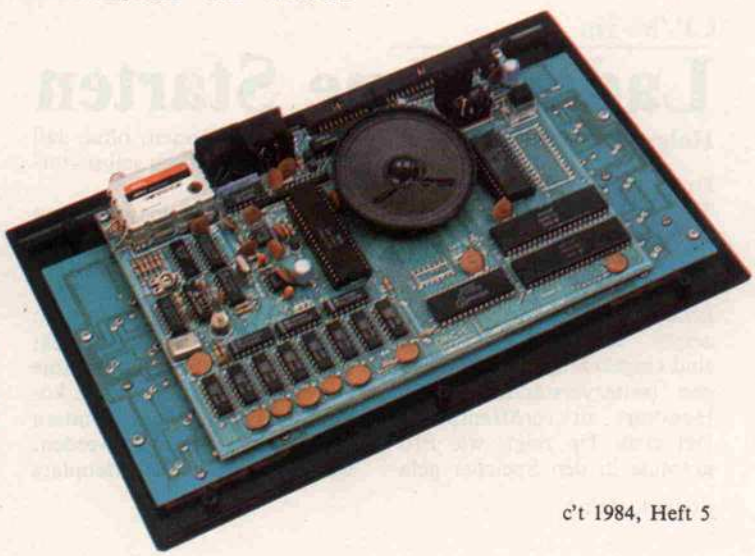
Das zum ATMOS mitgelieferte Demonstrationsprogramm bot für uns eine Überraschung: Der Computer zeichnete verschiedene Grafiken und spielte gleichzeitig eine Melodie. Dabei waren keine Unterbrechungen zu bemerken, obwohl die beiden Programme nicht wirklich gleichzeitig bearbeitet werden können. Stellt etwas das Betriebssystem des ATMOS Möglichkeiten zur 'quasi gleichzeitigen' (Interrupt-) Bearbeitung von Programmen bereit? Ein Blick in das Handbuch hilft da auch nicht weiter, da dort keine Angaben zu dieser doch reizvollen und wichtigen Betriebsart zu finden sind. Es bleibt also nur die Möglichkeit, das 'Demo'-Programm auf 'Tricks' zu durchsuchen, denn der Gedanke, ein Programm auszudrucken und gleichzeitig weiter daran zu schreiben, ist doch sehr verlockend.

In Farbe

Der ATMOS verfügt, wie der ORIC-1, über farbige hochauflösende Grafik mit 240 * 200 (Breite/Höhe) ansprechbaren Bildpunkten. Daneben gibt es noch eine einfache Blockgrafik, die mit der Schirmaufteilung des Text-Mode arbeitet (27 Zeilen zu 40 Zeichen). Bei der Text- und Grafikdarstellung können acht Farben verwendet werden: Schwarz, Rot, Grün, Gelb, Blau, Lila, Hellblau und Weiß. Alle Farben können gleichzeitig auf dem Schirm dargestellt werden.

Fazit

Der ORIC ATMOS ist kein neuer Computer, sondern eine begrüßenswerte Weiterentwicklung des ORIC-1. Dabei beseitigte man die Fehler und erleichterte die Bedienung des Gerätes. Erfreulich ist, daß der Hersteller offensichtlich keinen Wert auf futuristisches Äußeres oder die Verwendung einer 'werbewirksamen' 16-Bit-CPU legte. Der ORIC ATMOS — obwohl äußerlich klein — zählt zu den 'Größten' seiner Preisklasse. □



MTX-512: DER PROFI ZUM HOBBYPREIS

Schon wieder ein Neuer mögen Sie klagen und am liebsten umblättern. Aber an der MTX-SERIE kommen Sie nicht ungestraft vorbei. Technologisch von der Mittleren Datentechnik stammend, bietet die MTX-SERIE eine Vielzahl von Eigenschaften, die anderswo erst mit Software und Peripherie erkauft werden muß. So läßt sich schon das Grundgerät für die verschiedensten Spezialanwendungen einsetzen:

KAUFLEUTE werden die große Tastatur, den deutschen Zeichensatz, das numerische Tastenfeld, den Monitorausgang und den Centronics-Druckerausgang ebenso zu schätzen wissen, wie die Option, das System mit Diskettenstation unter CP/M® zu fahren (wem dann die maximale Speicherkapazität von 32 Megabyte Pseudofloppy und 20 Megabyte Winchester nicht genügt: bitte, dann müssen wir passen).

INGENIEUREN und Technikern dagegen imponiert da schon eher die eingebaute Echtzeituhr, der parallele I/O, der zugängliche Systembus, die umfassende technische Dokumentation sowie der im ROM implementierte ASSEMBLER/DISASSEMBLER (die Option mit 512 K RAM interessiert da schon weniger).

FREIZEITSPIELER, aber nicht nur diese, sondern auch Musiker und die eher auf Grafik erpichten Hobbyisten werden entzückt sein über die LOGO-gesteuerte Grafik, 32 Sprites mit 7 Parametern und dem 3-Kanal-HiFi-Tonausgang unter Softwaresteuerung (16 Farben und 2 Joystick Ports helfen natürlich auch weiter).

PROGRAMMIERER – professionell oder „just for fun" – werden die Gesamtheit aller Eigenschaften zu würdigen wissen, denn sie sind meist die Leidtragenden einer mangelhaft durchkonzipierten Maschine. Wer mit dem Befehl PANEL sofort alle Registerzustände, den Inhalt des RAMs und die Mnemonics des anstehenden Programms erfährt, weiß was er hat. Wer in einer BASIC-Zeile ASS. NNN eingibt, um anschließend eine ASSEMBLER-Routine zu schreiben, um dann in der Zeile NNN+1 in BASIC fortzufahren, ist schon ein Besserwisser (daß Sie dank des getrennten 16 K Bildschirmspeichers und dem selekt. 24 K ROM volle 64 K als Arbeitsspeicher zur Verfügung haben, soll hier nicht weiter stören).

ANFÄNGER, aber auch termingestreßte Programmierer können Memotech für das im ROM vorhandene NODDY danken. Mit nur sieben Befehlen lassen sich dialogorientierte Menüs, aber z.B. auch Bildschirmmasken ohne PRINT-Anweisung erstellen – kinderleicht (weshalb NODDY in England für den Anfangsunterricht eingesetzt werden soll).

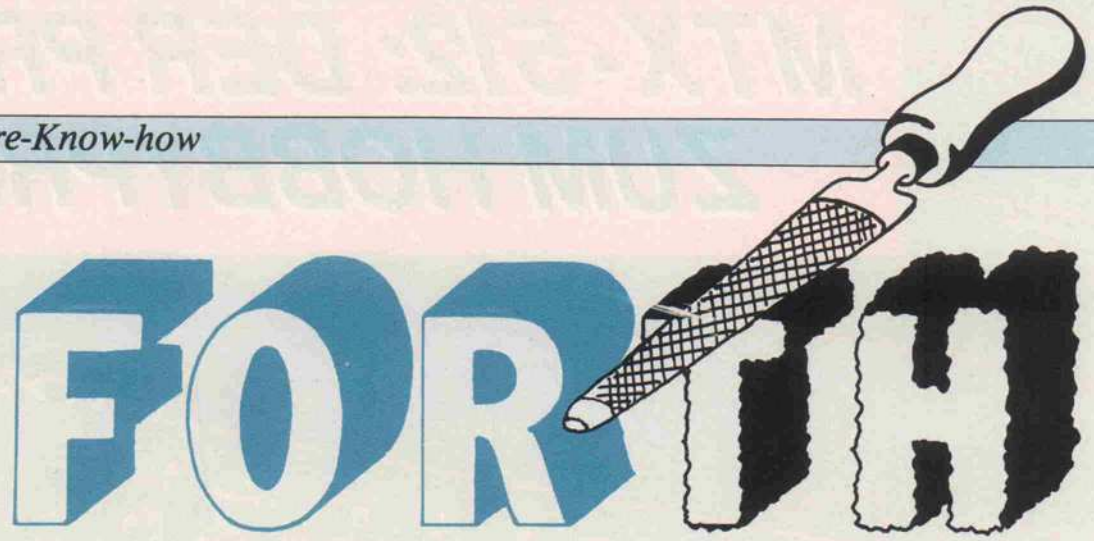
ALLE sollten wissen, daß die Memotech MTX-SERIE in 2 Grundversionen erhältlich ist (MTX-500/32 K RAM zu DM 1198,–; MTX-512/64 K RAM zu DM 1390,–), von Profisoft, dem „Software-Haus mit Service" vertrieben wird und Sie gegen Zusendung einer Postkarte weitere technische Unterlagen erhalten können. Selbstverständlich haben wir auch für den Fachhandel Unterlagen – mit anderem Schwerpunkt!

Memotech MTX-512 mit Diskettenstation

MEMOTECH MTX SERIES

profisoft

Sutthauser Str. 50–52 · 4500 Osnabrück · Tel. 05 41/5 39 05



anpassen

Teil 3: Der Massenspeicher

Peter Glasmacher

In den beiden vorausgegangenen Beiträgen wurde am Beispiel des APPLE II aufgezeigt, mit welchen Änderungen und Anpassungen man das FIG-FORTH mit einigem Bedienungskomfort lauffähig bekommt. Der dritte Teil der Serie beschäftigt sich nun mit einem der für den Bedienungskomfort wichtigsten Aspekte, nämlich der Anpassung der I/O-Routinen, um ein Diskettenlaufwerk betreiben zu können.

Disk I/0... aus dem Vollen geschnitzt

FORTH-Compiler, die ja eigentlich ein ganzes FORTH-System darstellen, sind für die verschiedensten Tischcomputer käuflich zu erwerben. Obwohl diese fertigen Systeme oftmals nicht wesentlich mehr als den FIG-FORTH-Standard bieten, liegt der Preis in den meisten Fällen über DM 200 und erreicht bei einigen Angeboten stolze DM 500. Der Preis eines FORTH-Systems resultiert meist aus der Tatsache, daß eben die Anpassung an einen Massenspeicher, zum Beispiel einer Diskettenstation, einiges an Systemprogrammiererfahrung erfordert, und diese Tätigkeit auch bezahlt werden muß. Da der Anbieter sein Unternehmen schließlich nach kaufmännischen Überlegungen führt, ergibt sich der manchmal etwas hobbyfeindliche Preis. Übrigens ein Grund für den Autor, ein FORTH zu entwerfen, welches dem Geldbeutel entgegenkommt und trotzdem komfortabel ist.

Als externen Speicher verlangt FORTH in der Regel ein Medium, welches wahlfreien Zugriff ermöglicht. Derzeit erfüllen lediglich Diskettenlaufwerke diese Bedingung zu einem einigermaßen annehmbaren Preis. Es wäre sicherlich möglich, auf Kassettenrecorder wahlfrei zuzugreifen, allerdings ist der Zeitbedarf dabei enorm. Deshalb soll diese Variante hier nicht diskutiert werden.

PREV	(--- addr)	Enthaelt die Adresse des zuletzt eingelesenen Puffers.
USE	(--- addr)	Enthaelt die Adresse des Puffers, in den der naechste Block eingelesen wird.
BLOCK	(n --- addr)	Liest den Block n in einen Puffer. Ist der Puffer durch einen geaenderten Block belegt, wird dieser Block zuerst zurueckgeschrieben.
BUFFER	(n --- addr)	Weist dem Block n einen Puffer zu. Ist der Puffer belegt, wird der Inhalt auf die Diskette geschrieben. Der Block wird NICHT eingelesen.
EMPTY-BUFFERS	(---)	Loescht alle Blockpuffer.
SAVE-BUFFERS	(---)	Schreibt alle geaenderten Puffer in den Massenspeicher.
LIST	(scr ---)	Listet einen Block mit ASCII Daten auf der Ausgabeeinheit.
LOAD	(scr ---)	Liest einen Block mit Quelldaten und fuehrt die in dem Block enthaltenen Anweisungen aus.
UPDATE	(---)	Markiert den aktuellen Blockpuffer (adressiert von PREV) als geaendert. Vor dem spaeteren Gebrauch des Puffers wird der Inhalt zurueckgeschrieben.

Tabelle 1. Das gesamte FORTH-Massenspeichergeschäft wird von relativ wenigen Worten unterstützt.

FORTH kennt kein Diskettenbetriebssystem (DOS) im eigentlichen Sinn, sondern stellt

Der FORTH-Massenspeicher

dem Anwender lediglich Basisworte zur Verfügung, mit denen sich Blocks mit einer festen Anzahl von Bytes, meist 1024, von der Platte in einen Puffer einlesen, manipulieren und wieder auf die Platte schreiben lassen (Tabelle 1). Es wird lediglich zwischen Einlesen (BLOCK) und Interpretieren (LOAD) unterschieden. Durch die Einteilung der Platte in feste Blöcke kann man auf jeden Block durch Angabe der Blocknummer zugreifen. Der vorbereitete Puffer enthält neben den Daten noch die Blocknummer selbst und ein Flag, welches den Zustand des Blocks signalisiert. Ist Bit 7 des Flags gesetzt, wird der dazugehörige Block immer auf die Platte zurückgeschrieben, bevor der entsprechende Puffer mit einem neuen Block belegt wird.

Diese relativ einfache Diskettenbehandlung erlaubt zum Beispiel, den verfügbaren Kernspeicher durch einfaches Neudefinieren von § und ! (FETCH und STORE) — natürlich unter Geschwindigkeitsverlust — um ein Vielfaches zu erweitern (Bild 1).

FORTH und die APPLE-DISK

Bei der Anpassung an die Diskettenhardware der APPLE-Rechner sind verschiedene Aspekte zu berücksichtigen. Die bei jedem Bootup geladene RWTS-Routine sollte ebenso wie der dazugehörige I/O-Block verwendet werden, um ein Kopieren einer FORTH-Diskette mit Programmen wie COPYA zu ermöglichen und

```
SCR #122
  Ø  ( NEUDEFINIERUNG $ , !                          Ø184GLASI)
  1 HEX
  2  : ?V<  DUP 8000 U< ;          ( ADRESSE < $8000 ?          )
  3  : VADRESS                     ( VIRTUELLE ADRESSE ERRECHN.)
  4     8000 - B/BUF /MOD          ( BLOCK UND OFFSET           )
  5     BC + BLOCK + ;             ( ADRESSE IM PUFFER          )
  6
  7  : $    ( addr --- n)
  8     ?V< IF $                   ( WENN < LADEN,              )
  9      ELSE VADRESS $ THEN ;     ( SONST BLOCK LADEN UND $    )
 10
 11  : !    ( n addr ---)
 12     ?V< IF !                   ( SPEICHERN                  )
 13      ELSE VADRESS ! UPDATE     ( ODER RECHNEN UND UPDATE    )
 14     THEN ;
 15  DECIMAL ;S
```

Bild 1. Durch Umdefinieren von Fetch und Store kann in FORTH ein virtueller Speicherzugriff realisiert werden.

ein Weiterverarbeiten von FORTH-Daten durch andere Programme nicht vollständig zu unterbinden. Bei der doch begrenzten Speicherkapazität von ca. 140 K einer APPLE-Disk sollte außerdem die einfache Möglichkeit bestehen, bei Bedarf ein zweites Laufwerk hinzuzufügen. Um bei der Arbeit mit mehreren Diskettenstationen Fehler auszuschließen, sollte auch eine einfache Fehlererkennung mit in die Massenspeicherroutinen eingebaut werden.

Das Ergebnis sind die Routinen in Bild 2, die erheblich vom vorgeschlagenen System im FIG-FORTH-Listing abweichen.

```
OK
123 TRIAD 126 LIST

SCR # 123
  Ø ( NEW R/W  1                                        Ø184glasi)
  1 -->
  2
  3 ØC48 CONSTANT BCNT                    ( Sektorzaehler        )
  4
  5 BDØØ CONSTANT RWTS                    ( APPLE RWTS Rtn.      )
  6
  7 B7EA CONSTANT IBDRVN                  ( Drivenummer          )
  8 B7EB CONSTANT IBVOL                   ( Volume               )
  9 B7EC CONSTANT IBTRK                   ( Tracknummer          )
 10 B7ED CONSTANT IBSECT                  ( Sektornummer         )
 11 B7FØ CONSTANT IBBUFP                  ( Pufferadresse        )
 12 B7F8 CONSTANT IOBPDN                  ( Vorh. Drivenr.       )
 13
 14
 15

SCR # 124
  Ø ( New R/W  2                                        Ø184glasi)
  1 HEX
  2
  3 CODE (R/W)
  4    XSAVE STX,                         ( X RETTEN             )
  5    4 # LDA,   ØC48 STA,               ( Sektorzaehler setzen )
  6    BEGIN,
  7       B7 # LDA,  EB # LDY,  BDØØ JSR,  ( Mit IOB-addr RWTS   )
  8       B7ED INC, B7ED LDA, ØF # AND,    ( Sekt.+1, neuer Track?)
  9         Ø= IF, B7EC INC, THEN,
 10       B7F1 INC, ØC48 DEC, Ø=           ( Buffer+1, cnt-1      )
 11    UNTIL, XSAVE LDX, NEXT JMP,         ( insgesamt 4 Sectoren )
 12 END-CODE
 13
 14 -->
 15

SCR # 125
  Ø ( New R/W  3                                        Ø184glasi)
  1
  2 : DERROR
  3    SWAP IF
  4       ."  ?DISC RANGE " EMPTY-BUFFERS ELSE
  5       ." *** WRITE PROTECTED  " QUIT THEN
  6    ELSE DROP THEN ;
  7
  8 : ?DRIVE
  9    B7EA C$ B7F8 C!  DFLAG $ IF         ( ZWEI LAUFWERKE ?     )
 10       OVER 88 > IF                    ( BLOCK >139     ?     )
 11          SWAP 8C - SWAP 2             ( LAUFWERK 2           )
 12       ELSE 1 THEN
 13       B7EA C!  THEN                   ( SONST DRIVE 1        )
 14    Ø B7EB C! ;                        ( VOLUME Ø             )
 15 -->
APPLEFORTH UA Ø9/83

SCR # 126
  Ø ( New R/W   4                                       Ø184glasi)
  1
  2 : R/W  ( buf blk flag ---)
  3    ?DRIVE                             ( Laufwerk feststellen )
  4       IF 1 ELSE 2 THEN                ( READ OR WRITE        )
  5       B7F4 C!
  6       DUP 88 > 1 DERROR               ( BLK >139 FEHLER !!   )
  7       4 10 */MOD                      ( TRACK UND SEKTOR
  8       B7EC C! B7ED C!                 ( --> IOB              )
  9       B7FØ !                          ( buf --> IBBUFP       )
 10          (R/W)                        ( OPERATION AUSFUEHREN )
 11       B7F5 C$ 10 =                    ( WRITE PROTECT ??     )
 12       Ø DERROR
 13 ;
 14 DECIMAL ;S
 15
OK
```

Bild 2. Die Diskettenroutinen unterstützen in der geänderten Form zwei Laufwerke und die Verarbeitung von 1024 Byte großen Blöcken.

Die gesamte Diskettenschnittstelle umfaßt nun drei Source Screens. Screen 123 ist als Referenz gedacht und erzeugt keinen Quellcode.

Zentraler Anlaufpunkt für alle Plattenoperationen ist nach wie vor das Wort R/W. Vor Aufruf von R/W müßen drei Parameter auf dem Stack übergeben werden:

Die Anfangsadresse des Pufferbereiches (buf), der dem zu bearbeitenden Block zugeordnet ist,

die Blocknummer selbst (blk) und das R/W-Flag mit einer 1 für eine Leseoperation beziehungsweise einer Ø für eine Schreiboperation (f).

R/W ruft als erstes ?DRIVE auf. Hier wird geprüft, wieviel Laufwerke vorhanden sind, der IOB wird entsprechend initialisiert, und bei zwei Laufwerken wird die Konstante 140 (ENTSPRICHT DER BLOCKANZAHL) von der Blocknummer subtrahiert.

Als zweiten Schritt setzt R/W den IOB auf eine Schreib- oder Leseoperation (Line 4) und verzweigt in eine Fehlerroutine, wenn die Blocknummer auf dem Stack trotz ?DRIVE größer als 139 ist (Line 6).

In den Zeilen 7 und 8 von Screen 126 werden aus der FORTH-Blocknummer die physikalischen Track/Sektor-Adressen errechnet und im IOB abgespeichert. Schließlich wird die Pufferadresse (buf) ebenfalls im IOB gespeichert. Damit sind alle nötigen Bedingungen erfüllt, und R/W ruft die einzige Maschinenroutine auf, die mit der Diskettenschnittstelle zusammenhängt. R/W liest vier aufeinanderfolgende Sektoren in den angegebenen Puffer, wobei das Erhöhen der Pufferadresse vom Maschinenprogramm durchgeführt wird.

Zuletzt sollte man noch einen kleinen aber wichtigen Trick in der Fehlerroutine beachten, der eigentlich der Ausgangspunkt für diese Routine war; ebenso hätte die FORTH-ERROR-Routine die Fehlerbehandlung übernehmen können.

Wird ein Range-Error, also eine zu hohe Blocknummer festgestellt, löscht DERROR lediglich alle Blockpuffer, um definierte Verhältnisse herzustellen, bei einem Write-Protect-Fehler verzweigt FORTH jedoch nach QUIT und bricht die Operation ab.

APPLE Spezifisches

Da jeder Tischcomputer seine Eigenheiten besitzt, existieren auch in jeder FORTH-Version spezielle, auf den Rechnertyp zugeschnittene Worte. Ein gutes Beispiel ist das FORTH des AIM 65, welches einen umfangreichen Wortschatz zur Behandlung der AIM Ein-/Ausgabegeräte und zur Kommunikation mit dem AIM-Editor enthält. Auch bei den APPLE-Rechnern wäre die Implementation einer Reihe von Worten denkbar, z. B. könnte man HIRES-Graphik-Befehle in FORTH integrieren. Naheliegender ist jedoch ein FORTH-Äquivalent zu den Worten PR # und IN # zu schaffen, um mit diesen Befehlen die Slots zu adressieren. Bild 3 zeigt den nötigen Code.

Damit wären die wichtigsten Routinen für die FORTH-Anpassung abgehandelt. Besitzer eines APPLE oder APPLE-artigen Rechners werden sicher einige konkrete Anregungen für verregnete Sommertage erhalten haben. Aber auch Leser, die keinen Rechner mit dem 6502 ihr eigen nennen, können nach diesem Schema vorgehen und ihr FORTH auf ähnliche Weise anpassen. □

```
SCR 130
  Ø ( APPLE PERIPHERIE                                  Ø184glasi)
  1 HEX
  2
  3 CODE IN#      ( n --- )
  4       XSAVE STX, TOP LDA,
  5         FE8B JSR,                  ( MONITOR ROUTINE      )
  6       XSAVE LDX, POP JMP,
  7 END-CODE
  8
  9 CODE PR#      ( n --- )
 10       XSAVE STX, TOP LDA,
 11       3 # CMP,  Ø= IF,             ( 80 ZEICHEN-KARTE ?   )
 12         PHA, 1 # LDA, D7CF LDY,    ( SFLAG SETZEN         )
 13         UP )Y STA, PLA,
 14       THEN, FE95  JSR,             ( MONITOR ROUTINE      )
 15       XSAVE LDX, POP JMP, END-CODE ;S
```

Bild 3. IN # und PR # in der FORTH-Form. Die Slotnummer wird auf dem Stack übergeben.

Herbert Nabereit

Assemblervirtuosen geraten bei der Nennung seines Namens in Verzückung, weniger versierte Programmierer verziehen säuerlich das Gesicht, möglicherweise stecken ihnen noch die letzten 132 Fatal Errors aus 145 Programmzeilen in den Gliedern. Diese unterschiedliche Reaktion ist gar nicht so verwunderlich: Die Vielfalt der Möglichkeiten, sich die Programmierarbeit zu erleichtern, geht einher mit der Vielfalt möglicher Fehler. Anhand selbst nachvollziehbarer Beispielprogramme zeigt dieser Beitrag, wie Sie den MACRO-80 dazu bringen können, daß er Sie für die unendliche Mühsal, die die Erstellung eines Assemblerprogrammes mit sich bringt, mit der Meldung 'No Fatal Error(s)' belohnt.

Teil 1

Macro 80 meistern

Ein Software-Werkzeug mit allen Schikanen

Die Programmierung von Mikrorechnern in höheren Sprachen findet ihre Grenzen vielfach bei Echtzeitanforderungen. Im Prinzip ist es möglich, Programme gleich im Maschinencode (HEX-Code) zu schreiben, die Mehrheit der Programmierer bevorzugt dazu aber ein Programm, das die mnemotechnische Darstellung der Assemblerbefehle in ein Maschinenprogramm umsetzt, dabei Adressen berechnet und diverse Hilfsarbeiten leistet, eben einen Assembler. Dies ist besonders dann von Bedeutung, wenn ein Programm später an andere Ein- oder Ausgabebausteine angepaßt werden muß beziehungsweise auf einer anderen Prozessorkarte (mit anderen Adressen) laufen soll.

Soll in den Code eines Compilers ein Maschinenprogramm eingebunden werden, so ist in der Regel ein entsprechender Assembler erforderlich, der an die Möglichkeiten dieses Compilers angepaßt ist. Der MACRO-80 ist ein Beispiel für einen überaus leistungsfähigen Assembler, der eine Reihe von Eigenschaften hat, die der Kompatibilität mit Compilern zugute kommen. So wird er beispielsweise beim Erwerb des Fortran-Compilers F80 der Firma Microsoft mitgeliefert.

Die große Zahl der Möglichkeiten beim Einsatz dieses Assemblers ist zugleich auch ein Nachteil, denn die Befehle müssen ja auch gelernt und behalten werden. Eine übersichtliche Darstellung der Befehle fehlt bisher, und nicht jeder ist auf englische Beschreibungen so eingestellt, daß er das Handbuch nach einmaligem Durchlesen verinnerlicht hat. Weiterhin gibt es etliche Sonderversionen für andere Betriebssysteme (z. B. für TRS-80 oder HDOS), die in Details von der CP/M-Version abweichen. Auch die Dokumentation ist von recht unterschiedlicher Qualität. Viele Anwender des MACRO-80 haben sich daher nur einen Grundstock von Befehlen zusammengesucht und vielleicht noch eine Beispieldatei erstellt, die als Muster für alle Programme benutzt wird.

Der MACRO-80 ist ein Softwarewerkzeug für den professionellen Anwender. Er kann fast alles, was sich ein Assemblerprogrammierer wünscht. Man muß nur wissen, wie man den MACRO-80 zu gewünschtem Tun überredet. Für die weitere Beschreibung setzen wir voraus, daß die Bedienung des CP/M-Betriebssystems bekannt ist und eine Diskette mit M80 und L80 bei ausreichendem freiem Platz im Drive A: steckt. Der gesamte Beitrag ist zum Mitarbeiten gedacht. Alle Programme sind auf einem CP/M-System lauffähig und erlauben dem Anwender, eigene Erfahrungen mit dem MACRO-80 zu bekommen.

Als erste Aufgabe sollen auf dem Terminal die Worte 'Hallo MACRO-80' ausgegeben werden. Das zugehörige Programm zeigt Beispiel 1.

Grundfunktionen

Um mit dem Assembler zu arbeiten, benötigt man zuerst eine Datei, die das Programm mit den mnemotechnischen Abkürzungen der 8080- oder Z80-Assemblersprache enthält. Das Beispielprogramm muß daher mit einem Editor wie WORDSTAR (unbedingt im NDOC-Mode!) oder ED eingegeben werden. Zur Verwendung mit dem MACRO-80 ist das Beispielprogramm unter einem Dateinamen mit der Extension .MAC auf der Diskette abzulegen, beispielsweise als M8001.MAC. Bei der Formulierung des Programms muß man sich an strenge Formatregeln halten, sonst liefert der MACRO-80 massenweise Fehlermeldungen. So müssen LABEL (= Symbolische Adressen) immer am Anfang der Zeile beginnen (ohne Leerzeichen) und einen Doppelpunkt am Ende haben. In einem LABEL sind die ersten sechs Zeichen signifikant, das heißt, nur sie werden vom MACRO-80 unterschieden. Das erste Zeichen muß ein Buchstabe sein. Beginnt ein hexadezimaler Operand mit einem Buchstaben, muß eine 0 vorgesetzt werden, weil der MACRO-80 den Operanden sonst als Symbol interpretiert.

```
LABEL: OPCODE OPERAND ; Kommentar
```

Zwischen LABEL, OPCODE und OPERAND reicht ein Leerzeichen (SPACE), ein optisch schöneres Bild erhält man durch die Verwendung von Tabulatorhaltepunkten (TAB). Kommentare müssen immer mit einem Semikolon (;) beginnen. Sie können auch am Anfang einer Zeile stehen.

Beispielprogramme

Zunächst wird im Beispielprogramm die Assemblersprache festgelegt, in der die folgenden Befehle geschrieben sind. Dies geschieht mit der Anweisung **.Z80** für die Z80-Assemblersprache (8080: **.8080**). Diese Anweisung kann innerhalb eines Programmes mehrfach neu gegeben werden, um zum Beispiel vorhandene 8080-Programme zusammen mit Z80-Programmen zu assemblieren. Hierzu ein Hinweis: Wenn Sie Ihre Programme mit dem WordStar erstellen, dürfen Textzeilen nicht in der ersten Spalte mit einem Punkt (.) beginnen, weil der WordStar diese Zeile dann als 'Dot-Befehl' auffaßt, also als Formatierungsanweisung. Auch wenn der WordStar diese Anweisung gar nicht ausführen kann, so unterschlägt er die ganze Zeile jedoch beim Ausdrucken. Andersherum 'meckert' der MACRO-80, wenn Sie Dot-Befehle des WordStar im Quelltext verwenden. Vor der MACRO-80-Anweisung '.Z80'

```
1:          TITLE   'MACRO-80-CONSOLE 01'
2:
3: ; Z80 - ASSEMBLERBEFEHLE BENUTZEN
4:
5:          .Z80
6:
7: ; ABSOLUTEN MASCHINENCODE ERZEUGEN
8:
9:          ASEG
10:
11: ; ERSTER BEFEHL AN DER STELLE 0200H IM SPEICHER
12:
13:         ORG     200H
14:
15: ; DEFINITION VON SYMBOLISCHEN KONSTANTEN
16:
17: SYSRES  EQU     0               ;SYSTEM-RESET BDOS-AUFRUF
18: CONOUT  EQU     2               ;CONSOLE-OUTPUT UEBER BDOS-AUFRUF
19: BDOS    EQU     5               ;EINSPRUNGADRESSE FUER BDOS-AUFRUF
20:
21:
22: ; HIER BEGINNT DAS PROGRAMM
23: ; ES SOLL 'HALLO MACRO-80' AUF DEM SCHIRM DES TERMINALS AUSGEGEBEN WERDEN
24:
25: START:  LD      HL,TEXT         ;AUF DEN TEXT-STRING ZEIGEN
26: OUTLOP: LD      A,(HL)          ;AKTUELLES ZEICHEN IN A LADEN
27:         BIT     7,A             ;IST BIT 7 DES A-REGISTERS GESETZT?
28:         JP      NZ,PENDE        ;JA, DANN IST DAS PROGRAMM ZUENDE
29:         LD      E,A             ;ZEICHEN FUER BDOS-AUFRUF LADEN
30:         PUSH    HL              ;HL-REGISTER VOR DEM BDOS RETTEN
31:         LD      C,CONOUT        ;CODE FUER BDOS-AUFRUF LADEN
32:         CALL    BDOS            ;BDOS DIE ARBEIT UEBERGEBEN
33:         POP     HL              ;TEXTZEIGER HOLEN
34:         INC     HL              ;AUF DAS NAECHSTE ZEICHEN ZEIGEN
35:         JP      OUTLOP          ;SCHLEIFE, BIS DER TEXT AUF DEM TERMINAL STEHT
36:
37: PENDE:  LD      C,SYSRES        ;SYSTEMRESET ALS BDOS-FUNKTION LADEN
38:         CALL    BDOS            ;BDOS GIBT DIE KONTROLLE AN CCP ZURUECK
39:
40: TEXT:   DEFB    'HALLO MACRO-80'  ;TEXT WIRD IN ASCII-ZEICHEN UMGESETZT
41:         DEFB    10,13,10,13,80H
42:
43:         END     START           ;PROGRAMMEINSPRUNG BEI START
```

Beispiel 1. 'HALLO MACRO-80' ausgeben

sollte also zumindest ein Leerzeichen oder ein TAB stehen, während Quelldateien erst mit Formatanweisungen verschönert werden dürfen, wenn sie nicht mehr assembliert werden müssen. Als nächstes wird die Art des zu erzeugenden Maschinencodes bestimmt. Die einfachste Art ist der absolute Maschinencode, das heißt, der Assembler bestimmt bei seinem Lauf die absoluten Adressen der Codes, eine spätere Verschiebung ist dann nicht mehr ohne weiteres möglich. Ein so erzeugtes Programm muß neu assembliert werden, wenn es an anderer Stelle im Adreßbereich eingesetzt werden soll. Die Erzeugung des absoluten Maschinencodes wird mit dem Befehl **ASEG** angeordnet. Eine Adressenvorgabe erfolgt mit dem Befehl **ORG XXXXH**, wobei XXXX die Adresse des folgenden Befehls im Hex-Code darstellt.

Jetzt erfolgt die Zuordnung von Werten zu den symbolischen Konstanten SYSRES, CONOUT und BDOS durch die **EQU**-Anweisung. Wird ein derart definiertes Symbol im Programm aufgerufen, so wird an seiner Stelle der mit EQU zugewiesene Wert übernommen (s. Beispiel 1, PENDE: DC, **SYSRES**). BDOS ist die Einsprungadresse zum Betriebssystem CP/M. Mit dem LABEL START: beginnt das Ausgabeprogramm. Adressen und erzeugter Maschinencode werden im Assemblerausdruck auf der linken Seite dargestellt. Der auszugebende Text steht als Operand des PSEUDO-OPCODES **DEFB**. Das letzte Byte des Textes erhält ein gesetztes 7. Bit (80 H). Auf diese Weise kann das Textende einfach erkannt werden. Am Programmende erwartet der MACRO-80 ein Ende-Kennzeichen, mit dem zugleich für das CP/M-Betriebssystem die Startadresse des Programmes festgelegt wird. Der Befehl **END XXXXH** kennzeichnet das Programmende, XXXXH die Einsprungadresse, die (wie im Beispiel) auch aus einer symbolischen Adresse (LABEL) bestehen kann.

Nachdem das Programm entsprechend dem Listing eingegeben wurde, muß es assembliert werden. Befindet sich der Assembler auf dem Laufwerk A:, so lautet der entsprechende Aufruf:

M80 (RETURN)

Der Assembler meldet sich dann mit einem '*' und erwartet die Eingabe eines Dateinamens:

*M8001, TTY: = M8001 (RETURN)

Dabei steht der erste Dateiname hinter dem '*' für die auszugebende Zwischencodedatei für den LINKer (der M80 versieht diese Datei mit der Extension .REL), der aus dieser Datei eine ausführbare Maschinencodedatei (Endung .COM) erzeugt. Die zweite 'Datei' bezeichnet den Dateinamen, unter dem das Listing abgespeichert werden soll. Es kann aber auch eine Ausgabe auf ein logisches Gerät angewiesen werden, zum Beispiel auf den Bildschirm (TTY:) oder den Drukker (LST:). Soll ein Listfile auf Diskette angelegt werden (.PRN), muß ein Dateiname ohne Doppelpunkt am Ende verwendet werden. Der Name der Quellendatei (= eingetipptes Programm) wird von der letzten Datei durch ein Gleichheitszeichen getrennt. Nachdem der MACRO-80 die Datei M8001.MAC assembliert hat, sollte die Meldung 'No Fatal Errors' erscheinen. Nun befindet sich das Listing des Programms auf dem Bildschirm und eine Datei mit dem Namen M8001.REL auf der Diskette. Diese Datei ist noch nicht ausführbar, da sie in einem speziellen internen Format erzeugt wurde, das auch noch Referenzen zu anderen .REL-Dateien enthalten kann. Beipielsweise wird dieses Format auch vom F80-FORTRAN-Compiler verwendet. Der LINKer arbeitet diese .REL-Datei in eine ausführbare Maschinencodedatei (.COM) um, die dann auf einem CP/M-System ablaufen kann. Der notwendige Aufruf dazu lautet:

L80 (RETURN)
*M8001 (RETURN)

Das Programm läßt sich mit dem folgenden Befehl auf der Diskette abspeichern.

*M8001/N/E

Dabei veranlaßt /N die Abspeicherung der .COM-Datei, und /E die Rückkehr ins Betriebssystem. Da es sich um ein Programm im Absolutmodus (ASEG) handelt, sind keine weiteren Eingaben nötig. Auf der Diskette findet sich eine Datei M8001.COM, die ausführbaren Maschinencode enthält. Mit dem Befehl M8001 (RETURN) kann das Programm nun ausgeführt werden. Es erscheint der Schriftzug 'HALLO MACRO-80' auf dem Terminal.

Operatoren und Assemblerdirektiven

Die mathematischen Operationen + — * / sind verwendbar, und es dürfen Klammern verwendet werden. Es gilt die Regel 'Punktrechnung vor Strichrechnung'. Die Zuordnung der Bearbeitungsreihenfolge ist in Tabelle 1 wiedergegeben. Diese Eigenschaft unterscheidet den MACRO-80 von vielen einfacheren Assemblern, die bei der einfachsten Mathematik schon in Schwierigkeiten geraten. Die Anwendung weiterer Operatoren zeigt das Beispielprogramm 2, in dem die Wirkung verschiedener Operatoren direkt am Maschinencode zu erkennen ist.

Reihenfolge der Bearbeitung arithmetischer Operatoren
NUL, TYPE
LOW, HIGH
*, /, MOD, SHR, SHL
Vorzeichen —
+, —
EQ, NE, LET, LE, GT, GE
NOT
AND
OR, XOR
Zu Beispiel 2: Im Ausdruck 3 * LOW 0F001H wird zunächst das LSB der 16-Bit-Zahl (01H) bestimmt und dann multipliziert.

Tabelle 1

Verwendung finden die Operatoren insbesondere bei adreßabhängigen Konstanten, aus denen bestimmte Werte abgeleitet werden müssen. Die Länge eines Speicherfeldes kann so als Differenz von Anfangs- und Endadressen bestimmt werden. Bei der Bestimmung des Inter-

ruptvektors für das I-Register des Z80 kann man zum Beispiel den HIGH-Befehl anwenden, um das MSB aus der Adresse der Interrupttabelle zu isolieren. Den Teil des Interruptvektors, den die PIO bekommt, isoliert man mit LOW.

Die logischen Operatoren (EQ, NE, LT, LE, GT, GE, NOT, AND, OR und XOR) erlauben den universellen Einsatz der bedingten Assemblierung, indem sie es gestatten, aus Adressen und Konstanten eine logische Bedingung zu bilden, die mit der IF-Anweisung abgefragt werden kann. Bei Verwendung der Operatoren ist zu beachten, daß nur +, —, * und / ohne Zwischenraum verwendet werden dürfen. In allen anderen Fällen müssen die Operatoren durch Zwischenräume von den Operanden getrennt sein.

Bedingte Assemblierung

Die Benutzung verschiedener Ein- oder Ausgabebausteine kann es erforderlich machen, daß je nach verwendeter Hardware unterschiedliche Ein- oder Ausgaberoutinen in einem Programm verwendet werden müssen. Zu diesem Zweck kann man natürlich verschiedene Programme benutzen, aber schon bei der Fehlerbeseitigung muß man dann jede Änderung doppelt ausführen. Eine bedingte Assemblierung von Programmteilen vermeidet diese doppelte Arbeit. Durch Setzen einer Konstanten kann man auf diese Weise festlegen, welche Teile eines Programmes assembliert werden sollen. Das Programm steht in einer Quelldatei und alle Änderungen sind jederzeit leicht verfügbar, ohne daß man sich besondere Gedanken um die Korrektur von mehreren Versionen machen muß.

Ein anderer Anwendungsbereich der bedingten Assemblierung ist die Fehlersuchphase (Debuggen) bei neu erstellten Programmen. Hier kann es hilfreich sein, gezielte Testausgaben in die Programme einzubauen, um den Programmablauf zu verfolgen. Dies gilt besonders bei Programmen, die etwa auf einem Einkartenprozessor ablaufen und keinen Bildschirm oder eine ähnliche Ein- beziehungsweise Ausgabe bedienen können. Ein Array aus mehreren Leuchtdioden

```
                                TITLE   FUNKTIONSDEMO

 0000'                          ASEG

                                .Z80

                        ; KONSTANTENDEFINITIONEN

 F00F                   ADRESS  EQU     0F00FH
 0097                   PIOCON  EQU     97H             ;HYPOTHETISCHE ADRESSE EINER PIO
 0400                   INTR1   EQU     400H            ;HYPOTHETISCHE INTERRUPTROUTINE
 0440                   INTR2   EQU     440H            ;HYPOTHETISCHE INTERRUPTROUTINE
 0480                   INTR3   EQU     480H            ;HYPOTHETISCHE INTERRUPTROUTINE


                        ; DIES PROGRAMM DEMONSTRIERT DIE FUNKTION DER M80-OPERATOREN
                        ; UND LIEFERT BEIM ABLAUF KEINE SINNVOLLE AUSGABE

                                ORG     0200H

                        ; SHIFTBEFEHLE FUER KONSTANTE: ACHTUNG ES WIRD MIT 0 AUFGEFUELLT

 0200   3E 02          START1: LD      A,1 SHL 1       ;00000001B SHL 1 = 00000010B 1 X LINKS
A 0202  3E 10                  LD      A,81H SHL 4     ;10000001B SHL 4 = 00010000B 4 X LINKS (!)
 0204   3E 40                  LD      A,80H SHR 1     ;10000000B SHR 1 = 01000000B 1 X RECHTS
 0206   3E 01                  LD      A,11H SHR 4     ;00010001B SHR 4 = 00000001B 4 X RECHTS(!)
 0208   21 00F0                LD      HL,0F00FH SHL 4 ;0F00FH SHL 4 = 00F0H
 020B   21 0F00                LD      HL,0F00FH SHR 4 ;0F00FH SHR 4 = 0F00H

                        ; DIE FEHLERMELDUNG A IN DER ZEILE HINTER START1 IST KORREKT, DENN ES
                        ; WURDE VERSUCHT IN A EINEN WERT ZU LADEN DER GROESSER ALS 255 IST

                        ; LSB ODER MSB-AUSWAHL BEI 16-BIT-KONSTANTEN (Z.B. ADRESSEN)

 020E   3E 0F                   LD      A,LOW ADRESS    ;LSB DER ADRESSE WIRD ISOLIERT
 0210   3E F0                   LD      A,HIGH ADRESS   ;MSB DER ADRESSE WIRD ISOLIERT

                        ; ARITHMETIK

 0212   3E FF                   LD      A,-1            ;ES WIRD 0 - ONE AUSGEFUEHRT
 0214   3E 30                   LD      A,10H+20H       ;BEIDE WERTE WERDEN ADDIERT
 0216   3E 20                   LD      A,30H-10H       ;TWO WIRD VON ONE ABGEZOGEN
 0218   3E 80                   LD      A,16*8          ;MULTIPLIKATION = 96
 021A   3E 0C                   LD      A,96/8          ;DIVISION OHNE REST = 16
 021C   3E 01                   LD      A,255 MOD 2     ;DISVISION DURCH 2, REST KOMMT IN A
 021E   21 0042                LD      HL,2+2*8*(2+2)  ;=66 PUNKTRECHNUNG VOR STRICHRECHNUNG

                        ; ISOLATION VON MSB UND LSB AUS EINER ADRESSE AM BEISPIEL DER
                        ; ERZEUGUNG DER INTERRUPTVEKTOREN DES Z80 UND DER Z80-PIO

 0221   3E 03                   LD      A,HIGH INTTAB   ;MSB DER ADRESSE DER INTERRUPTTABELLE ISOLIEREN
 0223   ED 47                   LD      I,A             ;IN Z80-INTERRUPTREGISTER LADEN
 0225   3E 12                   LD      A,LOW INTTAB    ;LSB DER ADRESSE DER INTERRUPTTABELLE ISOLIEREN
 0227   D3 97                   OUT     (PIOCON),A      ;IN DIE Z80-PIO LADEN

                                ORG     312H

                        ; DIE INTERRUPTTABELLE MUSS IMMER (!) AUF EINER GERADEN ADRESSE BEGINNEN

 0312   0400            INTTAB: DEFW    INTR1           ;ADRESSEN DER INTERRUPTROUTINEN 1..3
 0314   0440                    DEFW    INTR2
 0316   0480                    DEFW    INTR3

                        ; LOGISCHE OPERATOREN

 0318   FF                      DEFB    55 EQ 55        ;ZWEI WERTE SIND GLEICH  :WAHR
 0319   00                      DEFB    55.EQ 54        ;ZWEI WERTE SIND GLEICH  :FALSCH
 031A   00                      DEFB    55 NE 55        ;ZWEI WERTE SIND NICHT GLEICH  :FALSCH
 031B   FF                      DEFB    55 NE 54        ;ZWEI WERTE SIND NICHT GLEICH  :WAHR
 031C   FF                      DEFB    54 LT 55        ;ERSTER WERT IST KLEINER ALS DER ZWEITE :WAHR
```

```
031D  00                DEFB    55 LT 55        ;ERSTER WERT IST KLEINER ALS DER ZWEITE :FALSCH
031E  FF                DEFB    54 LE 55        ;ERSTER WERT KLEINER GLEICH DEM ZWEITEN :WAHR
031F  FF                DEFB    55 LE 55        ;ERSTER WERT KLEINER GLEICH DEM ZWEITEN :WAHR
0320  00                DEFB    55 LE 54        ;ERSTER WERT KLEINER GLEICH DEM ZWEITEN :FALSCH
0321  FF                DEFB    55 GT 54        ;ERSTER WERT IST GROESSER ALS DER ZWEITE :WAHR
0322  00                DEFB    54 GT 55        ;ERSTER WERT IST GROESSER ALS DER ZWEITE :FALSCH
0323  FF                DEFB    55 GE 54        ;ERSTER WERT GROESSER GLEICH DEM ZWEITEN :WAHR
0324  FF                DEFB    55 GE 55        ;ERSTER WERT GROESSER GLEICH DEM ZWEITEN :WAHR
0325  00                DEFB    55 GE 56        ;ERSTER WERT GROESSER GLEICH DEM ZWEITEN :FALSCH
0326  FF                DEFB    NOT 55 EQ 54    ;UMKEHRUNG DER LOGISCHEN AUSSAGE :WAHR
0327  FF                DEFB    55 EQ 55 AND 66 EQ 66   ; BEIDE BEDINGUNGEN TREFFEN ZU :WAHR
0328  00                DEFB    55 EQ 55 AND 66 EQ 64   ; BEIDE BEDINGUNGEN TREFFEN ZU :FALSCH
0329  FF                DEFB    55 EQ 55 OR 66 EQ 64    ; DIE EINE ODER DIE ANDERE : WAHR
032A  FF                DEFB    55 EQ 55 XOR 66 EQ 64   ; BEDINGUNGEN SIND VERSCHIEDEN : WAHR
032B  00                DEFB    55 EQ 55 XOR 66 EQ 66   ; BEDINGUNGEN SIND VERSCHIEDEN : FALSCH

                        END     START1

Macros:

Symbols:
ADRESS  F00F    INTR1   0400    INTR2   0440    INTR3   0480
INTTAB  0312    PIOCON  0097    START1  0200

1 Fatal error(s)
```

Beispiel 2. Die Wirkung der M-80-Operatoren

mit einem Treiber kann dann durch Hilfsroutinen angesprochen werden und dem Anwender angeben, welche Routine der Prozessor gerade abarbeitet. Bei vielen Anwendungen reicht es sogar, wenn gezielt Impulse auf einem Ausgang erzeugt werden, die man mit einem Oszilloscope identifizieren kann. Funktioniert das Programm, so kann man durch Änderung einer Konstanten alle Testroutinen aus dem Programm entfernen.

Die Grundform der bedingten Assembleranweisung ist

```
IF(BEDINGUNG) Ausdruck
...
(ELSE
... )
ENDIF
```

Erfüllt die Berechnung des Ausdrucks die Bedingung (TRUE), so wird der Text, der direkt hinter der IF-Anweisung steht, im Assemblerlauf berücksichtigt. Wird die Bedingung nicht erfüllt (FALSE), so wird der Text im Assemblerlauf verwandt, der hinter der ELSE-Anweisung bis zum ENDIF steht, falls überhaupt eine ELSE-Anweisung vorhanden ist. Die ENDIF-Anweisung ist immer nötig um das Ende des Bereiches der bedingten Assemblierung zu kennzeichnen. Anstelle von IF kann auch COND verwandt werden. Beispiele für die Anwendung der bedingten Assemblierung sind in den Beispielen 3 und 4 gegeben.

Im Beispiel 3 wird gezeigt, wie das Testprogramm von Beispiel 1 mit Hilfe der bedingten Assembleranweisungen wahlweise auf den Drucker oder den Bildschirm ausgegeben werden kann. Die Konstante PRINT steuert die Assemblierung des Programms, indem eine IF.. ELSE-Anweisung verwendet wird, die für PRINT = 0 den Terminalcode und bei PRINT = 1 den Druckercode an das BDOS übergibt.

Beispiel 4 demonstriert, wie mit Hilfe von bedingten Assemblierungen der Programmablauf auf dem Bildschirm dargestellt werden kann. Für jeden Zweig des Programms wird ein ASCII-Zeichen auf den Schirm ausgegeben. Auf diese Weise kann der Programmablauf verfolgt werden, auch wenn der Zweig wegen PRINT = 1 keine Ausgabe auf den Schirm erzeugen würde. Programmschleifen sind sofort zu erkennen und zu identifizieren. Nachdem das Programm lauffähig ist, wird DEBUG einfach zu 0 gesetzt, und der gesamte DEBUG-Code verschwindet aus dem Programm. Die Wirkung verschiedener 'vorgefertigter' IF-Anweisungen zeigt Beispiel 5. Der MACRO-80 enthält eine Reihe von IF-Anweisungen, die redundant sind. So kann zum Beispiel IFF (F = FALSE: assemblieren, wenn Bedingung nicht zutrifft) auch durch die Konstruktion IF/ELSE/ENDIF erzeugt werden. Bei der Arbeit mit Assemblercode dürfte die IF-Anweisung am häufigsten vorkommen (wie Beispiel 4).

Verschiebbare Programme

Der Maschinencode des 8080- und des Z80-Prozessors ist nicht ohne Änderungen im Speicherbereich verschiebbar. Im wesentlichen liegt das an der absoluten Adressierung der Sprünge und der Unterprogrammaufrufe. In der Regel sind die Codeteile, die zu absoluten Adressen weisen, bei ei-

```
1:          TITLE   'MACRO-80-CONSOLE/DRUCKER 03'
2:
3: ; Z80 - ASSEMBLERBEFEHLE BENUTZEN
4:
5:          .Z80
6:
7: ; ABSOLUTEN MASCHINENCODE ERZEUGEN
8:
9:          ASEG
10:
11: ; ERSTER BEFEHL AN DER STELLE 0200H IM SPEICHER
12:
13:         ORG     200H
14:
15: ; DEFINITION VON SYMBOLISCHEN KONSTANTEN
16:
17: SYSRES  EQU     0               ;SYSTEM-RESET BDOS-AUFRUF
18: CONOUT  EQU     2               ;CONSOLE-OUTPUT UEBER BDOS-AUFRUF
19: PRNOUT  EQU     5               ;DRUCKER-AUSGABE UEBER BDOS-AUFRUF
20: BDOS    EQU     5               ;EINSPRUNGADRESSE FUER BDOS-AUFRUF
21: PRINT   EQU     1               ;KENNZEICHEN FUER BEDINGTE ASSEMBLIERUNG
22:
23:
24: ; HIER BEGINNT DAS PROGRAMM
25: ; ES SOLL 'HALLO MACRO-80' AUF DEM SCHIRM DES TERMINALS
26: ; ODER AUF DEN DRUCKER, JE NACH DEM WERT VON PRINT AUSGEGEBEN WERDEN
27:
28: START:  LD      HL,TEXT         ;AUF DEN TEXT-STRING ZEIGEN
29: OUTLOP: LD      A,(HL)          ;AKTUELLES ZEICHEN IN A LADEN
30:         BIT     7,A             ;IST BIT 7 DES A-REGISTERS GESETZT?
31:         JP      NZ,PENDE        ;JA, DANN IST DAS PROGRAMM ZUENDE
32:         LD      E,A             ;ZEICHEN FUER BDOS-AUFRUF LADEN
33:         PUSH    HL              ;HL-REGISTER VOR DEM BDOS RETTEN
34:
35: ; HIER SOLL NUN PER BEDINGTER ASSEMBLERANWEISUNG DIE AUSGABE AUF DEN
36: ; DRUCKER ODER DEN BILDSCHIRM ERFOLGEN KOENNEN
37:
38:         IF      PRINT
39:         LD      C,PRNOUT
40:         ELSE
41:         LD      C,CONOUT        ;CODE FUER BDOS-AUFRUF LADEN
42:         ENDIF
43:
44:         CALL    BDOS            ;BDOS DIE ARBEIT UEBERGEBEN
45:         POP     HL              ;TEXTZEIGER HOLEN
46:         INC     HL              ;AUF DAS NAECHSTE ZEICHEN ZEIGEN
47:         JP      OUTLOP          ;SCHLEIFE, BIS DER TEXT AUF DEM TERMINAL STEHT
48:
49: PENDE:  LD      C,SYSRES        ;SYSTEMRESET ALS BDOS-FUNKTION LADEN
50:         CALL    BDOS            ;BDOS GIBT DIE KONTROLLE AN CCP ZURUECK
51:
52: TEXT:   DEFB    'HALLO MACRO-80'  ;TEXT WIRD IN ASCII-ZEICHEN UMGESETZT
53:         DEFB    10,13,10,13,80H
54:
55:         END     START           ;PROGRAMMEINSPRUNG BEI START
```

Beispiel 3. Durch IF-Anweisung Bildschirm oder Drucker ansprechen.

```
 1:        TITLE   'BEDINGTE ASSEMBLIERUNG #4'
 2:
 3:        .Z80
 4:        ASEG
 5:        ORG     2##H
 6:
 7: ; DEFINITION VON SYMBOLISCHEN KONSTANTEN
 8:
 9: SYSRES  EQU     #               ;SYSTEM-RESET BDOS-AUFRUF
10: CONOUT  EQU     2               ;CONSOLE-OUTPUT UEBER BDOS-AUFRUF
11: PRNOUT  EQU     5               ;DRUCKER-AUSGABE UEBER BDOS-AUFRUF
12: BDOS    EQU     5               ;EINSPRUNGADRESSE FUER BDOS-AUFRUF
13: PRINT   EQU     1               ;KENNZEICHEN FUER BEDINGTE ASSEMBLIERUNG
14: DEBUG   EQU     1               ;FLAG FUER DEBUGGERSCHLEIFE
15:
16:
17: ; ES SOLL AUF DEM BILDSCHIRM AUSGEGEBEN WERDEN, WIE WEIT DER ASSEMBLER IST
18:
19:        IF1
20:        .PRINTX *ERSTER ASSEMBLERDURCHLAUF*
21:        ENDIF
22:
23:        IF2
24:        .PRINTX *ZWEITER ASSEMBLERDURCHLAUF*
25:        ENDIF
26:
27: ; HIER BEGINNT DAS PROGRAMM
28: ; ES SOLL 'HALLO MACRO-8#' AUF DEM SCHIRM DES TERMINALS
29: ; ODER AUF DEN DRUCKER, JE NACH DEM WERT VON PRINT, AUSGEGEBEN WERDEN
30:
31: START:  LD      HL,TEXT         ;AUF DEN TEXT-STRING ZEIGEN
32:
33:        IF      DEBUG
34:        PUSH    HL
35:        LD      E,'A'           ;A FUER RICHTIGEN PROGRAMMEINSPRUNG
36:        LD      C,CONOUT        ;CODE FUER BDOS-AUFRUF LADEN
37:        CALL    BDOS            ;BDOS DIE ARBEIT UEBERGEBEN
38:        POP     HL
39:        ENDIF
40:
41: OUTLOP: LD     A,(HL)          ;AKTUELLES ZEICHEN IN A LADEN
42:        BIT     7,A             ;IST BIT 7 DES A-REGISTERS GESETZT?
43:        JP      NZ,PENDE        ;JA, DANN IST DAS PROGRAMM ZUENDE
44:        LD      E,A             ;ZEICHEN FUER BDOS-AUFRUF LADEN
45:        PUSH    HL              ;HL-REGISTER VOR DEM BDOS RETTEN
46:
47: ; HIER SOLL NUN PER BEDINGTER ASSEMBLERANWEISUNG DIE AUSGABE AUF DEN
48: ; DRUCKER ODER DEN BILDSCHIRM ERFOLGEN KOENNEN
49:
50:        IF      PRINT
51:
52: ; MARKIERUNG VON VERSCHIEDENEN OPTIONEN DURCH ANDERE KENNZIFFERN
53: ; INNERHALB EINER BEDINGTEN ASSEMBLERANWEISUNG
54:
55:        IF      DEBUG
56:        PUSH    DE              ;ZEICHEN RETTEN
57:        PUSH    HL
58:        LD      E,'P'           ;ZIFFER P FUER DEN DRUCKERAUFRUF
59:        LD      C,CONOUT        ;CODE FUER BDOS-AUFRUF LADEN
60:        CALL    BDOS            ;BDOS DIE ARBEIT UEBERGEBEN
61:        POP     HL
62:        POP     DE
63:        ENDIF
64:
65:        LD      C,PRNOUT
66:        ELSE
67:
68:        IF      DEBUG
69:        PUSH    DE              ;ZEICHEN RETTEN
70:        PUSH    HL
71:        LD      E,'T'           ;ZIFFER T FUER DAS TERMINAL
72:        LD      C,CONOUT        ;CODE FUER BDOS-AUFRUF LADEN
73:        CALL    BDOS            ;BDOS DIE ARBEIT UEBERGEBEN
74:        POP     HL
75:        POP     DE
76:        ENDIF
77:
78:        LD      C,CONOUT        ;CODE FUER BDOS-AUFRUF LADEN
79:        ENDIF
80:
81:        CALL    BDOS            ;BDOS DIE ARBEIT UEBERGEBEN
82:        POP     HL              ;TEXTZEIGER HOLEN
83:        INC     HL              ;AUF DAS NAECHSTE ZEICHEN ZEIGEN
84:        JP      OUTLOP          ;SCHLEIFE, BIS DER TEXT AUF DEM TERMINAL STEHT
85:
86: PENDE:  LD      C,SYSRES        ;SYSTEMRESET ALS BDOS-FUNKTION LADEN
87:        CALL    BDOS            ;BDOS GIBT DIE KONTROLLE AN CCP ZURUECK
88:
89: TEXT:   DEFB    'HALLO MACRO-8#'  ;TEXT WIRD IN ASCII-ZEICHEN UMGESETZT
90:        DEFB    1#,13,1#,13,8#H
91:
92:        END     START           ;PROGRAMMEINSPRUNG BEI START
```

Beispiel 4. Hilfsroutinen per IF-Anweisung ein-/ausschalten

```
 1:        TITLE   'BEDINGTE ASSEMBLIERUNG #5'
 2:
 3:        .Z80
 4:        ASEG
 5:        ORG     2##H
 6:
 7: ; DEFINITION VON SYMBOLISCHEN KONSTANTEN
 8:
 9: DEBUG   EQU     1               ;FLAG FUER DEBUGGERSCHLEIFE
10:
11: ; DIE VERWENDUNG DIESER ASSEMBLIERUNGSBEDINGUNGEN
12: ; BIETET SICH SPEZIELL BEI MACRO'S AN. AUF DIESE WEISE
13: ; KANN REAGIERT WERDEN, WENN EIN MACRO FALSCH EINGESETZT WIRD
14: ; MIT HILFE DES .PRINTX-BEFEHLS KANN EIN HINWEIS AUF DEN FEHLER
15: ; AUF DEN TERMINAL AUSGEGEBEN WERDEN, ES IST ABER AUCH MOEGLICH
16: ; IN DIESEM FALL EIN ENTSPRECHENDES PROGRAMM AUTOMATISCH
17: ; EINZUBINDEN
18:
19:        IFDEF   DEBUG
20:        .PRINTX * DEBUG-MODE WIRD ASSEMBLIERT *
21:        ENDIF
22:
23:        IFNDEF  DEBUG
24:        .PRINTX * DEBUG-KONSTANTE NICHT DEFINIERT *
25:        ENDIF
26:
27:        IFB     <>
28:        .PRINTX * DAS ARGUMENT IST NICHT ANGEGEBEN *
29:        ENDIF
30:
31:        IFNB    <XYZ>
32:        .PRINTX * ES IST EIN ARGUMENT VORHANDEN *
33:        ENDIF
34:
35:        IFIDN <ABCDE>,<ABCDE>
36:        .PRINTX * DIE ARGUMENTE SIND GLEICH *
37:        ENDIF
38:
39:        IFIDN <ABCDE>,<EDCBA>
40:        .PRINTX * DIE ARGUMENTE SIND NICHT GLEICH *
41:        ENDIF
42:
43:        END     2##H
```

Beispiel 5. Die Wirkung diverser M80-eigener IF-Befehle

ner Verschiebung des Maschinencodes zu korrigieren. Eine automatische Korrektur ist aber nicht möglich, da bekannt sein muß, welche Daten einen Bezug zum Programm haben und welche Daten unabhängig vom Programm sind (zum Beispiel Aufrufe von Programmen aus einem Monitor, I/O-Adressen, Konstanten).

Der bisher verwendete Absolut-Modus (ASEG) des Assemblers erlaubt die Bearbeitung von Assemblerprogrammen, wobei immer der gesamte Quellencode vom Assembler verarbeitet wird. Soll ein Programm später auf einem EPROM oder ROM abgespeichert werden, so wird schon im Quellencode ein Datenbereich festgelegt, der an einer Adresse im RAM-Bereich liegt. Das hat aber zur Folge, daß Speicherzellen, die von Unterprogrammen verwendet werden, auch alle in diesem Speicherbereich liegen müssen und direkt im Code des Unterprogramms nicht sichtbar sind.

Will man die entsprechenden Unterprogramme abspeichern und eventuell als komplette Datei mit INCLUDE in das Programm übernehmen, so müssen

die Speicherplätze jedesmal wieder in den gemeinsamen Speicherbereich eingetragen werden. Teilt man beim Schreiben des Programmes schon in Code und Speicher auf und assembliert so, daß beide Bereiche getrennt sind, dann erhält man am Ende des Assemblerlaufes zwei Blöcke. Ein Block enthält den Maschinencode (Code-Segment), der zweite die Speicherplätze (Daten-Segment). Durch diese Aufteilung ist dann bekannt, welche LABEL oder SYMBOLE sich auf Adressen beziehen und welche unabhängig von den Programmadressen sind. LABEL oder symbolische Konstanten, die sich auf die Programmadressen beziehen, lassen sich während des Assemblerlaufes noch nicht endgültig festlegen, da ihr Wert erst dann definiert werden kann, wenn die tatsächlichen Adressen bekannt sind. Der Assembler erzeugt daher eine Datei, in der diese Werte gekennzeichnet sind und zunächst relativ zum vorgegebenen Programmbeginn (ORG) liegen. Dies ist der sogenannte relative Code (.REL-Datei). Neben den noch nicht endgültig festgelegten Adressen bietet der MACRO-80 die Möglichkeit, Symbole mit im REL-Code abzulegen, deren Wert erst später vom LINKer zugeordnet wird. Diese Möglichkeit erlaubt es, verschiedene .REL-Module zu koppeln. Die Zuordnung der endgültigen Adressen erfolgt im LINKer, einem Programm, das .REL-Dateien übernimmt und entsprechend bestimmter Vorgaben die Programm- und die Datenadressen festlegt.Nun ist ein ablauffähiges Maschinenprogramm entstanden, dessen Codeteil auf ein EPROM abgespeichert werden kann.

Ein Beispiel für die Arbeitsweise geben die Beispiele 6, 7 und 8, die zusammen ein einfaches Programm ergeben, das von der Tastatur eingegebene Zeichen zwischenspeichert und erst bei einem Carriage-Return/ Linefeed (CR/LF) wieder ausgibt. Mit EXTRN müssen in jedem Programm die LABEL definiert werden, die das Programm zwar aufruft, die jedoch nicht im aktuellen Modul stehen. Im Programmlisting 6 sind dies OUTSTR, CBUFFER und PRINTT. Gleichzeitig müssen mit PUBLIC anderen Programmodulen die Adressen zugänglich gemacht werden, die für diese Module EXTRN sind (in Listing 6: ZEIGER). Gleiches ist auch in den anderen Programmodulen erforderlich.

In allen Listings findet sich der Befehl INCLUDE FNAME, der eine andere Art der Modulbildung erlaubt. Mittels INCLUDE können beliebige Dateien mit assembliert werden. Diese Art der Modulbildung erlaubt es, häufig verwendete Programme als Quellencode in ein Assemblerprogramm einzubinden, ohne diese Programme vorher in einer Datei zusammenzufassen. Im vorliegenden Fall wird diese Möglichkeit genutzt, um die Systemkonstanten in jedem der drei Programmodule zu verwenden. Das Programm aus den Listings 6..8 kann auf diese Weise auch durch einen Assemblerlauf umgesetzt werden, dabei entfallen dann die EXTERN und PUBLIC-Definitionen. Listing 9 zeigt dieses Beispiel.

```
 1:        TITLE   PROGRAMMBEISPIEL FUER CSEG-ANWENDUNG #6
 2:
 3:        .Z80
 4:        CSEG
 5:        ORG     #H
 6:
 7:        INCLUDE M8SYSDE.MAC
 8:
 9: ; HIER BEGINNT DAS PROGRAMM
10: ; ES MELDET SICH MIT DER BOTSCHAFT' PROGRAMM 6' AUF DEM SCHIRM DES TERMINALS
11:
12:        PUBLIC  ZEIGER
13:        EXTRN   OUTSTR,CBUFFER,PRINTT   ;NAMEN IN ANDEREN .REL-DATEIEN
14:
15: START: LD      HL,TEXTB        ;AUF DEN TEXT-STRING ZEIGEN
16:        CALL    OUTSTR          ;TEXT AUF DEN SCHIRM AUSGEBEN
17:        LD      HL,CBUFFER
18:        LD      (ZEIGER),HL
19: PLOOP:
20:        LD      HL,TEXTE
21:        CALL    OUTSTR          ;AUFFORDERUNG ZUR EINGABE
22:        CALL    GETCHR          ;ZEICHEN VON DER CONSOLE HOLEN
23:        CP      '#'             ;BEI # WIRD ABGEBROCHEN!
24:        JP      Z,PENDE
25:        CP      CR              ;BEI CR WIRD DAS 7.BIT ZU H GESETZT
26:                                ;UND DIE EINGABE AUF DEM BILDSCHIRM
27:                                ;WIEDERGEGEBEN
28:        JP      Z,AUSGAB
29:        CALL    STORE
30:        JP      PLOOP           ;NAECHSTES ZEICHEN HOLEN
31:
32: AUSGAB: SET    7,A             ;7.BIT SETZEN
33:        CALL    STORE
34:        CALL    PRINTT          ;STRING AUF DEM TERMINAL AUSGEBEN, ZEIGER
35:        JP      PLOOP           ;WIRD WIEDER AUF DEN ANFANG GESETZT
36:
37: PENDE: LD      C,SYSRES        ;SYSTEMRESET ALS BDOS-FUNKTION LADEN
38:        CALL    BDOS            ;BDOS GIBT DIE KONTROLLE AN CCP ZURUECK
39:
40: ; ZEICHEN IM BUFFER DES UNTERPROGRAMMS ABLEGEN
41:
42: STORE: LD      HL,(ZEIGER)     ;ZEIGER ZUM NAECHSTEN SPEICHERPLATZ
43:        LD      (HL),A          ;ZEICHEN ABLEGEN
44:        INC     HL
45:        LD      (ZEIGER),HL
46:        RET
47:
48: ; ZEICHEN VON DER TASTATUR DES TERMINALS HOLEN
49:
50: GETCHR: LD     C,CONIN         ;BDOS-AUFRUF
51:        CALL    BDOS
52:        RET                     ;DAS ZEICHEN STEHT NUN IM A-REGISTER
53:
54: TEXTB:  DEFB   'PROGRAMM 6'     ;TEXT WIRD IN ASCII-ZEICHEN UMGESETZT
55:        DEFB    CR,LF
56:        DEFB    'PROGRAMMENDE MIT #'
57:        DEFB    CR,LF,CR,LF,#0H
58: TEXTE:  DEFB   CR,LF
59:        DC      'BITTE BELIEBIGE TASTE DRUECKEN: '
60:
61:        DSEG
62: ZEIGER: DS     2               ;2 BYTE FUER BUFFERZEIGER IM UNTERPROGRAMM
63:
64:
65:        END     START           ;PROGRAMMEINSPRUNG BEI START
```

Beispiel 6. Die Beispiele 6, 7 und 8 zeigen verschiebbare Module

```
 1:        TITLE   'RELOKALISIERBARE MODULE #7 - STRINGAUSGABE'
 2:
 3:        .Z80
 4:        CSEG
 5:        ORG     #H
 6:
 7:        INCLUDE M8SYSDE.MAC
 8:
 9: ; HIER BEGINNT DAS PROGRAMM
10: ; ES SOLL DER TEXT AUF DEM SCHIRM DES TERMINALS AUSGEGEBEN WERDEN,
11: ; AUF DESSEN ERSTES ZEICHEN DAS HL-REGISTER ZEIGT. DER TEXT ENDET
12: ; MIT EINEM GESETZTEN 7.BIT, DAS LETZTE ZEICHEN WIRD MIT AUSGEGEBEN
13: ; R: A,C,E,HL: HL BEI EINSPRUNG GESETZT
14:
15:        PUBLIC  OUTSTR          ;OUTSTR FUER ANDERE PROGRAMM ZUGAENGLICH
16:                                ;MACHEN
17:
18: OUTSTR: LD     A,(HL)          ;AKTUELLES ZEICHEN IN A LADEN
19:        LD      E,A             ;ZEICHEN FUER BDOS-AUFRUF LADEN
20:        PUSH    AF              ;AKTELLES ZEICHEN RETTEN
21:        PUSH    HL              ;HL-REGISTER VOR DEM BDOS RETTEN
22:        LD      C,CONOUT        ;CODE FUER BDOS-AUFRUF LADEN
23:        CALL    BDOS            ;BDOS DIE ARBEIT UEBERGEBEN
24:        POP     HL
25:        POP     AF
26:        BIT     7,A             ;IST BIT 7 DES A-REGISTERS GESETZT?
27:        RET     NZ              ;JA, DANN IST DAS PROGRAMM ZUENDE(RETURN)
28:        INC     HL              ;AUF DAS NAECHSTE ZEICHEN ZEIGEN
29:        JP      OUTSTR          ;SCHLEIFE, BIS DER TEXT AUF DEM TERMINAL STEHT
30:
31:        END
```

Beispiel 7.

```
1: ; MODUL ZUR AUSGABE EINES TEXTSTRINGS UND ZUR ADRESSE KORREKTUR
2:
3:          .Z80
4:          CSEG
5:          ORG     00H
6:
7:          INCLUDE M8SYSDE.MAC
8:
9:          PUBLIC  PRINTT,CBUFFE
10:         EXTRN   ZEIGER,OUTSTR
11:
12: PRINTT:
13:         CALL    LFCR
14:         CALL    LFCR
15:         LD      HL,CBUFFE       ;ANFANG DES BUFFERS
16:         CALL    OUTSTR          ;STRINGS AUSGEBEN
17:         LD      HL,CBUFFE
18:         LD      (ZEIGER),HL
19:         CALL    LFCR
20:         RET
21:
22: LFCR:   LD      HL,LFCRT
23:         CALL    OUTSTR
24:         RET
25:
26: LFCRT:  DEFB    LF,CR+80H
27:
28:         DSEG
29:
30: CBUFFE: DEFS    200             ;BUFFER FUER EINGABE
31:
32:         END
```

```
1:
2: ; DEFINITION VON SYMBOLISCHEN KONSTANTEN
3:
4: SYSRES  EQU     0               ;SYSTEM-RESET BDOS-AUFRUF
5: CONIN   EQU     1               ;SYSTEMCONSOLE LESEN
6: CONOUT  EQU     2               ;CONSOLE-OUTPUT UEBER BDOS-AUFRUF
7: PRNOUT  EQU     5               ;DRUCKER-AUSGABE UEBER BDOS-AUFRUF
8: BDOS    EQU     5               ;EINSPRUNGADRESSE FUER BDOS-AUFRUF
9: PRINT   EQU     1               ;KENNZEICHEN FUER BEDINGTE ASSEMBLIERUNG
10: DEBUG  EQU     1               ;FLAG FUER DEBUGGERSCHLEIFE
11:
12: ; ASCII-KONSTANTEN
13:
14: CR     EQU     13
15: LF     EQU     10
```

Beispiel 8. Die Module Beispiel 6—8 werden einzeln assembliert, benutzen aber gemeinsame Konstanten aus M8SYSDE.MAC (unterer Teil Bsp. 8)

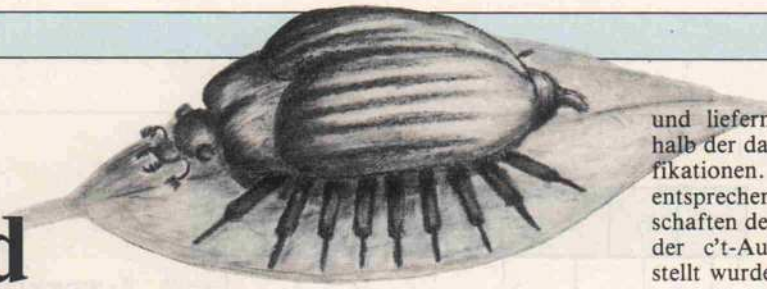
```
1:          TITLE   PROGRAMMBEISPIEL FUER CSEG-ANWENDUNG 09
2:
3: ; PROGRAMMMODULE WERDEN DURCH INCLUDE GEKOPPELT, EXTERN UND PUBLIC-
4: ; DEFINITIONEN ENTFALLEN, ALLE DATEN UND BEFEHLE SIND RELATIV ZUM
5: ; ORG16IN (ORG 00H) DEFINIERT
6:
7:          .Z80
8:          CSEG
9:          ORG     00H
10:
11:         INCLUDE M8SYSDE.MAC
12:
13: ; HIER BEGINNT DAS PROGRAMM
14: ; ES MELDET SICH MIT DER BOTSCHAFT' PROGRAMM 6' AUF DEM SCHIRM DES TERMINALS
15:
16: START:  LD      HL,TEXTB        ;AUF DEN TEXT-STRING ZEIGEN
17:         CALL    OUTSTR          ;TEXT AUF DEN SCHIRM AUSGEBEN
18:         LD      HL,CBUFFER
19:         LD      (ZEIGER),HL
20: PLOOP:
21:         LD      HL,TEXTE
22:         CALL    OUTSTR          ;AUFFORDERUNG ZUR EINGABE
23:         CALL    GETCHR          ;ZEICHEN VON DER CONSOLE HOLEN
24:         CP      '#'             ;BEI # WIRD ABGEBROCHEN!
25:         JP      Z,PENDE
26:         CP      CR              ;BEI CR WIRD DAS 7.BIT ZU H GESETZT
27:                                 ;UND DIE EINGABE AUF DEM BILDSCHIRM
28:                                 ;WIEDERGEGEBEN
29:         JP      Z,AUSGAB
30:         CALL    STORE
31:         JP      PLOOP           ;NAECHSTES ZEICHEN HOLEN
32:
33: AUSGAB: SET     7,A             ;7.BIT SETZEN
34:         CALL    STORE
35:         CALL    PRINTT          ;STRING AUF DEM TERMINAL AUSGEBEN, ZEIGER
36:         JP      PLOOP           ;WIRD WIEDER AUF DEN ANFANG GESETZT
37:
38: PENDE:  LD      C,SYSRES        ;SYSTEMRESET ALS BDOS-FUNKTION LADEN
39:         CALL    BDOS            ;BDOS GIBT DIE KONTROLLE AN CCP ZURUECK
40:
41: ; ZEICHEN IM BUFFER DES UNTERPROGRAMMS ABLEGEN
42:
43: STORE:  LD      HL,(ZEIGER)     ;ZEIGER ZUM NAECHSTEN SPEICHERPLATZ
44:         LD      (HL),A          ;ZEICHEN ABLEGEN
45:         INC     HL
46:         LD      (ZEIGER),HL
47:         RET
48:
49: ; ZEICHEN VON DER TASTATUR DES TERMINALS HOLEN
50:
51: GETCHR: LD      C,CONIN         ;BDOS-AUFRUF
52:         CALL    BDOS
53:         RET                     ;DAS ZEICHEN STEHT NUN IM A-REGISTER
54:
55: TEXTB:  DEFB    'PROGRAMM 6'    ;TEXT WIRD IN ASCII-ZEICHEN UMGESETZT
56:         DEFB    CR,LF
57:         DEFB    'PROGRAMMENDE MIT #'
58:         DEFB    CR,LF,CR,LF,80H
59: TEXTE:  DEFB    CR,LF
60:         DC      'BITTE BELIEBIGE TASTE DRUECKEN: '
61:
62:         DSEG
63: ZEIGER: DS      2               ;2 BYTE FUER BUFFERZEIGER IM UNTERPROGRAMM
64:
65:         INCLUDE M8009B.MAC
66:         INCLUDE M8009C.MAC
67:
68:         END     START           ;PROGRAMMEINSPRUNG BEI START
```

```
1:          CSEG
2:
3: ; HIER BEGINNT DAS PROGRAMM
4: ; ES SOLL DER TEXT AUF DEM SCHIRM DES TERMINALS AUSGEGEBEN WERDEN,
5: ; AUF DESSEN ERSTES ZEICHEN DAS HL-REGISTER ZEIGT. DER TEXT ENDET
6: ; MIT EINEM GESETZTEN 7.BIT, DAS LETZTE ZEICHEN WIRD MIT AUSGEGEBEN
7: ; R: A,C,E,HL: HL BEI EINSPRUNG GESETZT
8:
9:
10: OUTSTR: LD     A,(HL)          ;AKTUELLES ZEICHEN IN A LADEN
11:         LD      E,A             ;ZEICHEN FUER BDOS-AUFRUF LADEN
12:         PUSH    AF              ;AKTELLES ZEICHEN RETTEN
13:         PUSH    HL              ;HL-REGISTER VOR DEM BDOS RETTEN
14:         LD      C,CONOUT        ;CODE FUER BDOS-AUFRUF LADEN
15:         CALL    BDOS            ;BDOS DIE ARBEIT UEBERGEBEN
16:         POP     HL
17:         POP     AF
18:         BIT     7,A             ;IST BIT 7 DES A-REGISTERS GESETZT?
19:         RET     NZ              ;JA, DANN IST DAS PROGRAMM ZUENDE(RETURN)
20:         INC     HL              ;AUF DAS NAECHSTE ZEICHEN ZEIGEN
21:         JP      OUTSTR          ;SCHLEIFE, BIS DER TEXT AUF DEM TERMINAL STEHT
```

```
1:          CSEG
2:
3: PRINTT:
4:          CALL    LFCR
5:          CALL    LFCR
6:          LD      HL,CBUFFE       ;ANFANG DES BUFFERS
7:          CALL    OUTSTR          ;STRINGS AUSGEBEN
8:          LD      HL,CBUFFE
9:          LD      (ZEIGER),HL
10:         CALL    LFCR
11:         RET
12:
13: LFCR:   LD      HL,LFCRT
14:         CALL    OUTSTR
15:         RET
16:
17: LFCRT:  DEFB    LF,CR+80H
18:
19:         DSEG
20:
21: CBUFFE: DEFS    200             ;BUFFER FUER EINGABE
```

Beispiel 9. Die Programme 6—8 werden zusammengefaßt assembliert. Systemvariablen werden aus M8SYSDE übernommen.

Im 2. Teil wird der Umgang mit MACROs und dem Linker beleuchtet. Dazu gibt es eine Liste der wichtigsten Befehle. □



VIA 6522: Einsatz und Programmierung

Eckart Steffens

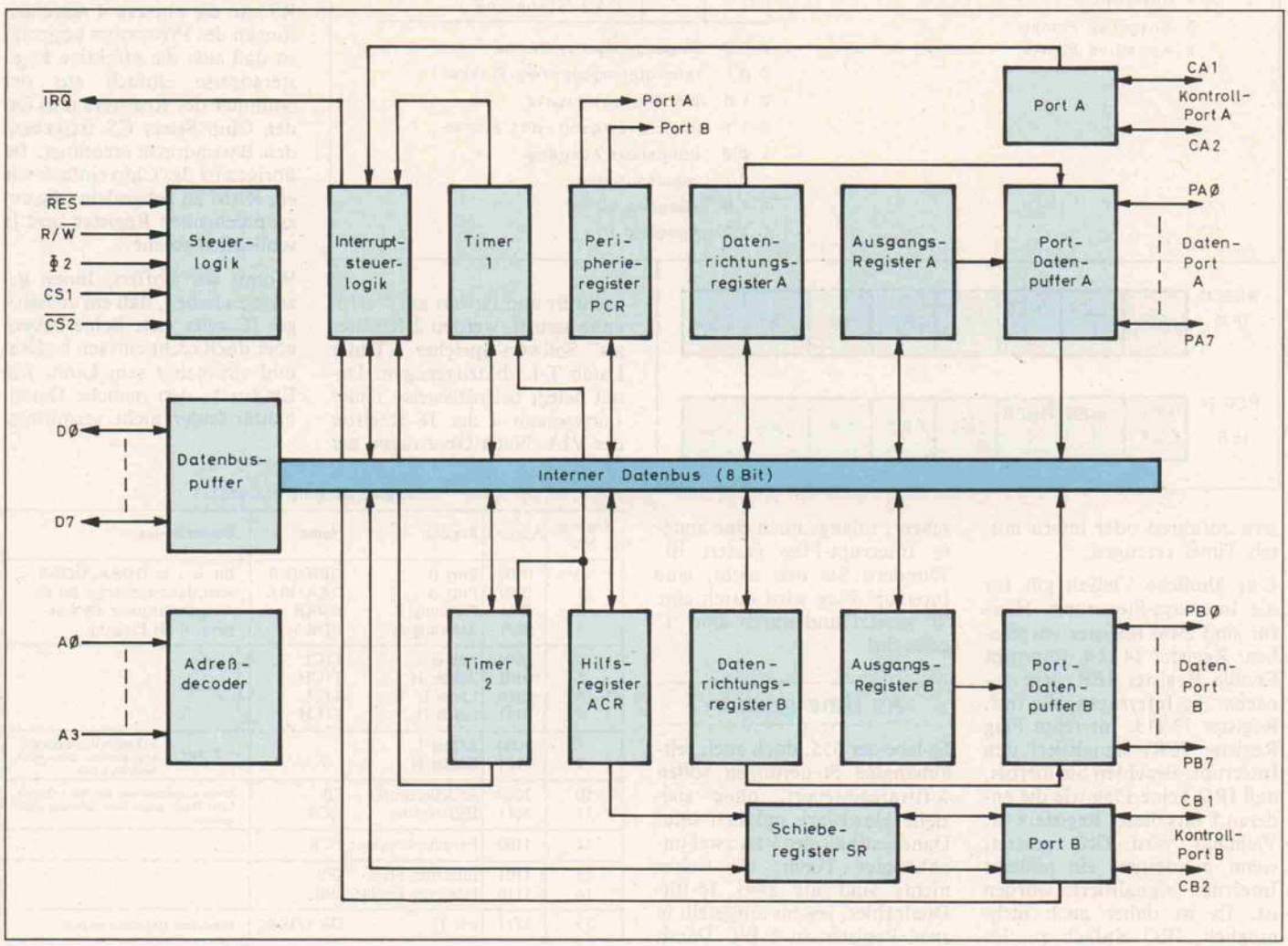
Einer der vielseitigsten (und wohl meistverbreiteten: ORIC, CBM, ACORN, ...) I/O-Bausteine ist der vielseitige Interface-Adapter 6522, von verschiedenen Herstellern angeboten und in Kürze auch als CMOS-Version G65SC22 zu haben. Was sich hinter diesem 40beinigen 'Maikäfer' verbirgt und wie man ihn einsetzt, soll im folgenden Gegenstand unserer Betrachtung sein.

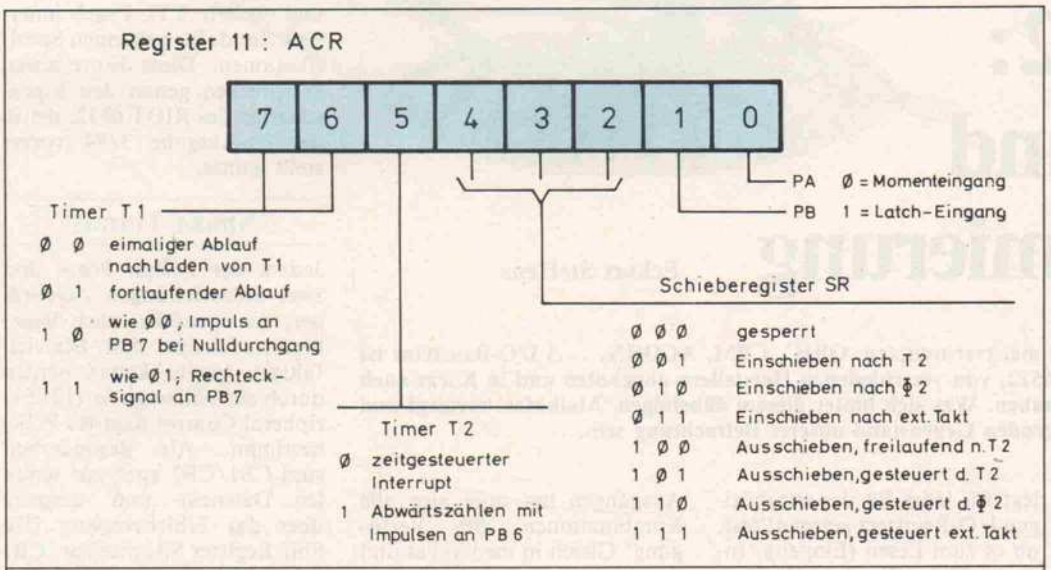
16 In/Out-Leitungen, zwei Ports zu je 8 Bit Breite, können getrennt bedient werden: Zwei der 16 Register des 6522, genannt A und B, bewältigen den Verkehr mit der Außenwelt (1: Input/Output-Register B IRB/ORB, 2: Input/Output-Register A IRA/ORA). In welcher Richtung, entscheidet ein jeweils zugehöriges Richtungsregister (2, 3: DataDirectionRegister DDRA, DDRB). Dieses legt für jedes Bit des zugehörigen I/O-Registers getrennt fest, ob es zum Lesen (Eingang, InputRegister IR) oder Schreiben (Ausgang, OutputRegister OR) verwendet werden soll. Die Stellen der Richtungsregister und Datenregister korrespondieren zueinander. Eine 1 in Bit 3 von DDRB bedeutet also, daß Bit 3 des I/ORB als Ausgang geschaltet ist.

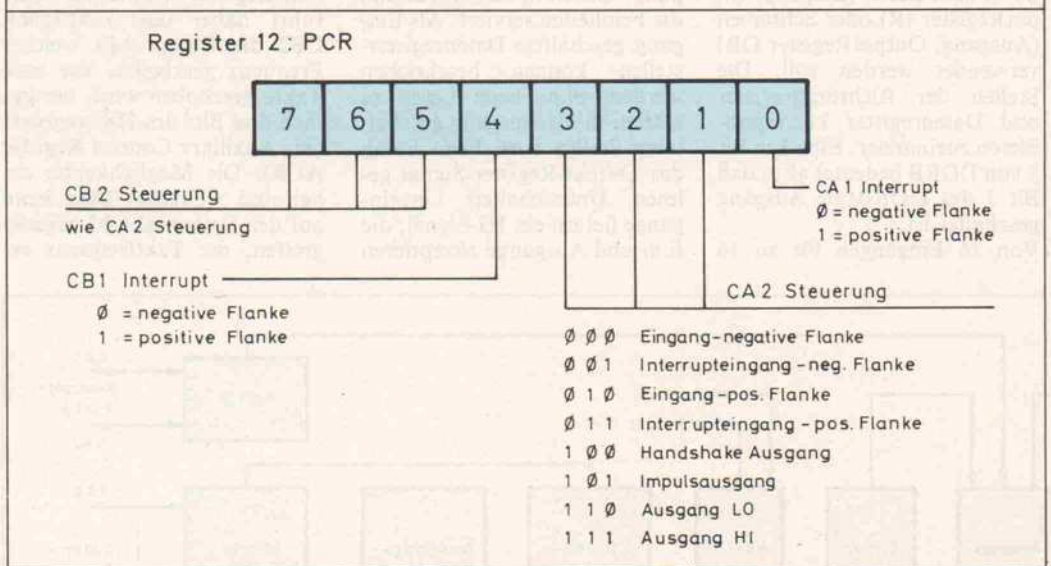
Von 16 Eingängen bis zu 16 Ausgängen hat man also alle Kombinationen zur Verfügung! Gleich in medias res und die Feinheiten serviert: Als Eingang geschaltete Datenregisterstellen können beschrieben werden, ohne beim Lesen zu stören; in als Ausgang geschalteten Stellen wird beim Lesen das Output-Register-Signal gelesen. Unbeschaltete Leseeingänge liefern ein HI-Signal; die Ein- und Ausgänge akzeptieren und liefern TTL-Pegel innerhalb der dafür geltenden Spezifikationen. Diese Einzelheiten entsprechen genau den Eigenschaften des RIOT 6532, der in der c't-Ausgabe 3/84 vorgestellt wurde.

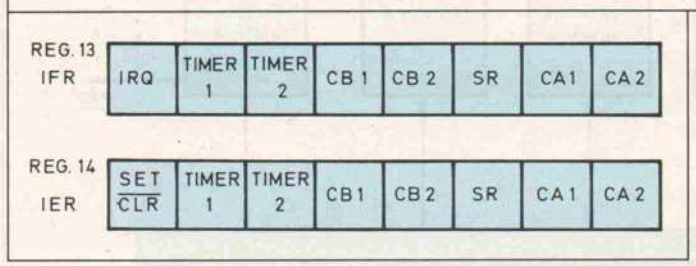
Shake Hands

Jedem der beiden Ports sind zwei Steuerleitungen zugeordnet, die ebenfalls 'nach Belieben' zu schalten sind. Die vielfältigen Möglichkeiten werden durch ein Hilfsregister (12: Peripheral Control Register PCR) bestimmt. Als Besonderheit sind CB1/CB2 auch zur seriellen Datenein- und -ausgabe über das Schieberegister (10: ShiftRegister SR) nutzbar. CB1 führt dabei das Taktsignal, CB2 die Daten. Mit welcher Frequenz gearbeitet, wie viele Takte geschoben wird, bestimmen drei Bits des Hilfsregisters (11: Auxiliary Control Register ACR). Die Möglichkeiten dabei sind vielfältig: Man kann auf den Systemtakt Φ2 zurückgreifen, die Taktfrequenz ex-









tern zuführen oder intern mittels Timer erzeugen.

Eine ähnliche Vielfalt gilt für die Interrupt-Steuerung. Hierfür sind zwei Register vorgesehen: Register 14 (14: Interrupt Enable Register IER) legt zunächst die Interruptquelle fest, Register 13 (13: Interrupt Flag Register IFR) signalisiert den Interrupt. Beachten Sie hierbei, daß IRQ keine Flag wie die anderen 7 Bits dieses Registers ist. Vielmehr wird IRQ gesetzt, wenn mindestens ein anderer Interrupt signalisiert worden ist. Es ist daher auch nicht möglich, IRQ einfach zu 'löschen', solange noch eine andere Interrupt-Flag gesetzt ist. Wundern Sie sich nicht: eine Interrupt-Flag wird durch eine '0' gesetzt und durch eine '1' gelöscht!

As time goes by

Es lebe der 555, doch auch zeitabhängige Steuerungen sollen softwaregesteuert, ohne spezielle Hardware möglich sein. Daher enthält der VIA zwei unabhängige Timer, die weiter nichts sind als zwei 16-Bit-Dualzähler, jeweils aufgeteilt in zwei Register zu 8 Bit. Damit Sollwert und Istwert zur Verfügung stehen, werden 2 Register als Sollwert-Speicher (Timer Latch T-L) hinzugezogen. Damit belegt beispielsweise Timer 1 insgesamt 4 der 16 Register des VIA. Nach Übertragen der Daten aus dem Latch in den Zähler zählt dieser mit dem Systemtakt Φ2 abwärts, bis er 0 erreicht, setzt dann seine Interrupt-Flag im IFR und stoppt (es sei denn, es ist über die Steuerbits 6 und 7 im ACR eine andere Programmierung getroffen). Sind diese beiden Bits gesetzt, wiederholt sich der Ablauf ständig und bei jedem 'Nulldurchgang' (TimeOut) wird PB7 invertiert. Timer 1 erzeugt damit ein Rechtecksignal mit einer durch die Latches T1L und T1H einstellbaren Frequenz. Timer 2 ist ebenfalls 16 Bit breit, belegt jedoch nur 2 Register und verfügt daher nur über etwas eingeschränktere Möglichkeiten, die durch das Steuerbit 5 im ACR gewählt werden.

Adressierung

Damit sind die wichtigsten Eigenschaften des VIA benannt. Seine Stärke liegt in der Vielfalt der Möglichkeiten und der Einfachheit der Programmierung. Im Regelfall wird man die Registeradreßleitungen RS0... RS3 an die unteren 4 Adreßleitungen des Prozessors koppeln, so daß sich die effektive Registeradresse einfach aus der Nummer des Registers plus der den Chip Select CS freigebenden Basisadresse errechnet. Im übrigen ist der Chip einfach wie ein RAM zu behandeln: die anzusprechenden Register sind ja schließlich solche.

Womit wir hoffen, Ihnen gezeigt zu haben, daß ein vielseitiges IC zwar viele Beine haben, aber doch recht einfach bedien- und einsetzbar sein kann. Ein Eindruck, den manche Datenblätter leider nicht vermitteln. □

Registerbelegung VIA 6522

Register Nr.	Adresse	Register	Name	Beschreibung
0	0000	Port B	ORB/IRB	Bit = 1 in DDRA/DDRB setzt das zugehörige Bit im Ausgangsregister als Ausgang, 0 als Eingang
1	0001	Port A	ORA/IRA	
2	0010	Richtung B	DDRB	
3	0011	Richtung A	DDRA	
4	0100	Zähler L	T1CL	Timer 1
5	0101	Zähler H	T1CH	
6	0110	Latch L	T1LL	
7	0111	Latch H	T1LH	
8	1000	Zähler L		Timer 2 Es kann in Reg. 8 geschrieben werden. Dies wirkt dann als Latch.
9	1001	Zähler H		
10	1010	Schieberegister	SR	Erstes ausgeschobenes Bit: Bit 7 (Rotate Left) Daten gehen beim Schieben nicht verloren.
11	1011	Hilfsregister	ACR	
12	1100	Peripherieregister	PCR	
13	1101	Interrupt Flags	IFR	
14	1110	Interrupt Enable	IER	
15	1111	wie 1)	ORA/IRA	jedoch kein Handshake möglich

c't-Floppy-Interface in ECB-Systemen

Kurt Werner

Das in c't 2/84 vorgestellte Floppy-Disk-Interface wurde zwar für den 16-Bit-Computer c't86 entwickelt, ist aber für (fast) alle ECB-Bus-Computer verwendbar. In diesem Artikel zeigen wir, welche Änderungen an der Karte vorgenommen werden müssen und wie die Software zur Ansteuerung aussehen könnte. Das Floppy-Disk-Interface zeichnet sich durch seine hohe Nachbausicherheit und einen einfach vorzunehmenden Abgleich aus.

Modifikationen

Alle von uns vorgeschlagenen Modifikationen können, müssen aber nicht vorgenommen werden. Die Floppy-Karte ist ohne Änderungen an Z80-Rechnern lauffähig; allerdings ist die '16-Bit'-Version teurer als die 'abgemagerte' Z80-Ausführung.

Es können folgende Modifikationen ausgeführt werden: IC 13 sowie R4 und C9 entfallen ersatzlos. Die 'Jumperfelder' BR3, BR4, BR5 und BR6 haben für den Betrieb am Z80 keine Bedeutung mehr und brauchen nicht bestückt zu werden. Ebenfalls entfallen kann der DIP-Schalter S, da die Software alle Decodierungen ausführen muß. Die Steckerleiste X1 sollte man durch eine Ausführung mit 64 Polen ersetzen.

Software

Soll die Floppy-Karte an einem CP/M-System betrieben werden, müssen alle Programmteile, die die Disketten Ein-/Ausgabe betreffen, an den Controller angepaßt werden. Die zu ändernden Routinen stehen im BIOS, dem einzigen hardwareabhängigen Teil des CP/M. Die hier abgedruckten Routinen sollen zeigen, wie die Änderungen für ein CP/M 2.2 im Prinzip aussehen. Die Programme sind also als Beispiele aufzufassen. Ein komplettes BIOS-Listing können wir nicht abdrucken, weil für jeden Computertyp eine spezielle Anpassung erforderlich ist.

Die Routinen, die sich auf das Floppy-Disk-Interface beziehen, sind neben den Boot-Programmen: HOME, SELECT DISK, SET TRACK, SET SEKTOR, SET DMA-ADRESSE, READ SEKTOR und WRITE SEKTOR. Im folgenden beschreiben wir die Funktion dieser einzelnen Routinen.

HOME

Mit diesem Programm wird der Kopf des Laufwerks über der Spur Ø der Diskette positioniert. Dazu muß an den Controller ØCh ausgegeben werden. Zu dem eigentlichen Controller-Kommando muß noch die Step-Rate addiert werden. Die Werte für STEP sind der Tabelle 1 zu entnehmen.

Tabelle 1

Step-Rate Bit 1	Step-Rate Bit Ø	Zeit in ms 5″	Zeit in ms 8″
Ø	Ø	6	3
Ø	1	12	6
1	Ø	2Ø	1Ø
1	1	3Ø	15

Einstellbare Step-Rate

SELECT DISK

Diese Routine gibt an den Floppy-Controller die Laufwerksnummer aus. Dabei bestimmen die oberen Bits des ausgegebenen Bytes noch einige Funktionen des Controllers. Die Funktion der einzelnen Bits sind in der Tabelle 2 angegeben. Zusätzlich berechnet das Programm noch die Adresse der Disk-Parameter-Tabelle, die der beim CP/M-86 entspricht. Für jedes Laufwerk muß eine Routine SELECT DISK geschrieben werden.

Tabelle 2

Bit	Funktion
7	5¼″-Drive 'Motor-on'
6	Start Timer (entfällt)
5	Bit 'Ø' = Double Density Bit '1' = Single Density
4	Bit 'Ø' = 5¼″-Drive Bit '1' = 8″-Drive
3	Drive-Select 4
2	Drive-Select 3
1	Drive-Select 2
Ø	Drive Select 1

Funktionen des Controllers

SET TRACK

SET TRACK positioniert den Floppy-Kopf über einer bestimmten Spur. Dazu wird das Datenregister des Controllers mit der Nummer der gewünschten Spur geladen. Anschließend wird an das Controll-Register 1Ch + Steprate ausgegeben.

SET SEKTOR

Mit dieser Routine wird bestimmt, welcher Sektor zu lesen oder zu beschreiben ist. Dazu wird das Sektorregister des Controllers mit der gewünschten Sektornummer geladen.

SET DMA-ADRESSE

Mit SET DMA-ADRESSE wird angegeben, wohin Daten 'transportiert' werden sollen. Dazu muß lediglich die 'DMAADR' abgespeichert werden.

READ SEKTOR

Das ist die eigentliche Lese-Routine. READ-SEKTOR liest in einer Schleife den eingestellten Sektor. Ist dieser Sektor größer als 128 Byte (5¼″, Double Density), so muß zusätzlich noch ein Blocking-/Deblocking-Algorithmus verwendet werden, wie er im CP/M Alteration Guide beschrieben ist.

WRITE SEKTOR

Mit WRITE SEKTOR wird der eingestellte Sektor der Diskette beschrieben. Diese Routine ist dem Programm 'READ SEKTOR' sehr ähnlich.

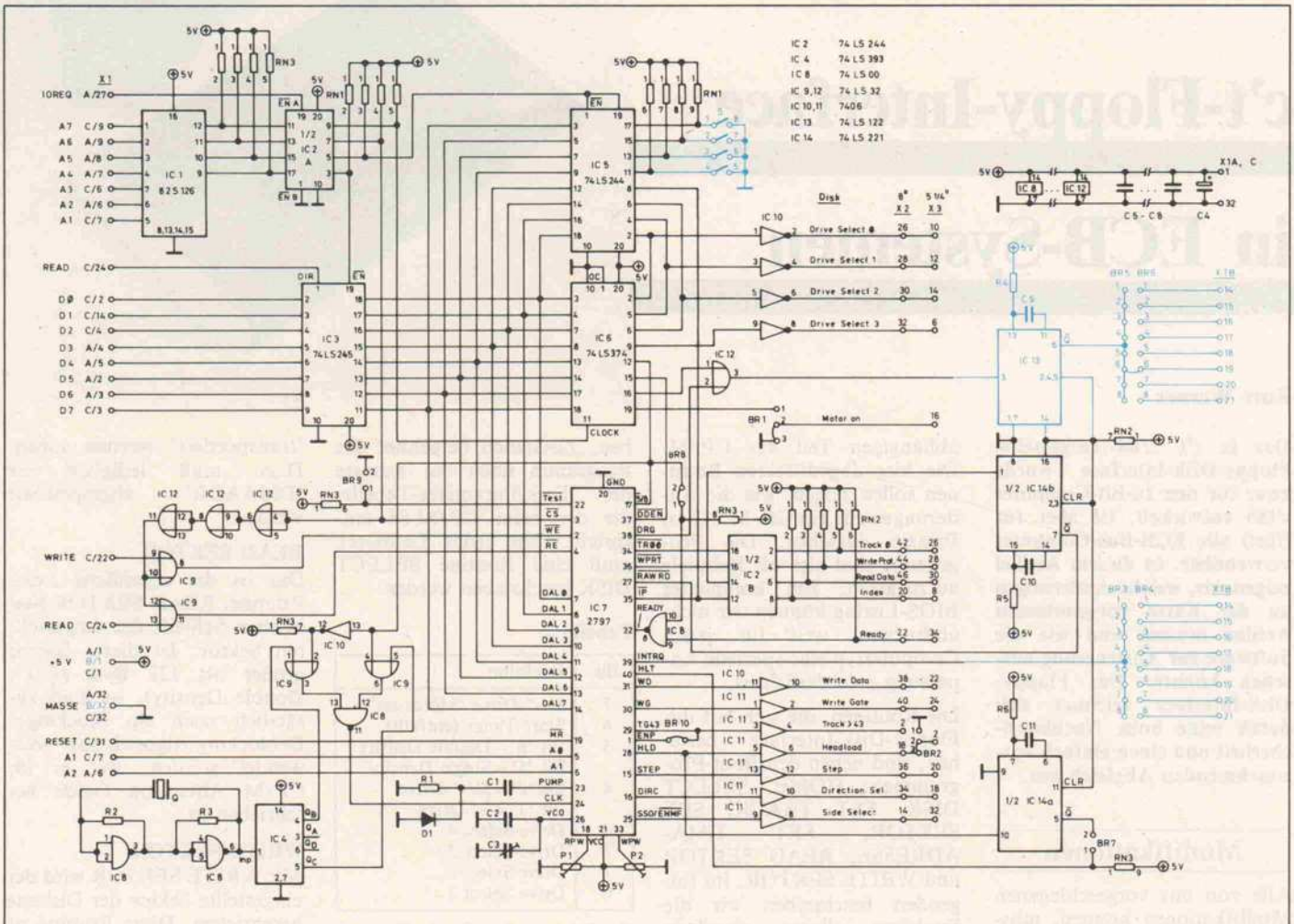
Booten

Die weiteren Änderungen am BIOS betreffen die Warmboot-Routine. Dieses Programm muß die ersten beiden Spuren der Diskette lesen, in denen das Betriebssystem steht.

Der 'Bootstrap-Loader' ist ebenfalls an den Controller anzupassen. Dieser Programmteil dient dazu, das Betriebssystem nach dem Einschalten (oder Reset) von der Diskette in den Speicher zu laden. 'Warmboot' und 'Bootstrap-Loader' lassen sich nach dem Muster der 'READ-SEKTOR'-Routine erstellen.

Befehl	Funktion
ØC + Steprate	Kopf auf Spur Ø positionieren
1C + Steprate	Spur suchen
88 + Side	Sektor lesen
A8 + Side	Sektor schreiben
CØ + Side	Lesen der aktuellen Spur
DØ	Floppy-Controller zurücksetzen
Side Ø ≙ Seite Ø 2 ≙ Seite 1	

Eine vereinfachte Darstellung der Kommandos des 2797



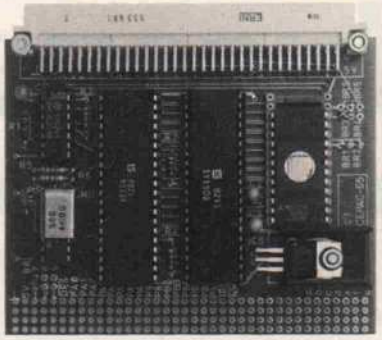

```
;
STEP    EQU     1           ;Werte dem Drive entspr.
                            ;einsetzen
;
FUNKT   EQU     0           ;Hier die dem Drive entspr.
                            ;Werte nach Tab.2 einsetzen
;
;
;
;Controller-Adressen
;
FLOPC   EQU     80H         ;Controll-Register
FLOPT   EQU     82H         ;Track-Register
FLOPS   EQU     84H         ;Sektor-Register
FLOPD   EQU     86H         ;Data-Register
FLOPO   EQU     88H         ;Drive-Format-Register
;
;
;HOME
;
HOME:   LD      A,0CH+STEP  ;Akku mit HOME-Kommando
        OUT     (FLOPC),A   ;laden und ausgeben
        CALL    WAIT        ;auf Controller warten
        RET
;
;
;SELECT DISK
;
SELECT: LD      HL,0        ;Bei Fehler ist HL=0
        LD      A,C         ;CP/M übergibt Drive-Nummer in C
        CP      1           ;erstes Drive gewählt?
        RET     NC          ;Nein-Abbruch
        AND     0FH         ;Vordere Bits löschen (sie
        OR      FUNKT       ;bestimmen die Drive-Nummer)
        OUT     (FLOPO),A   ;Drive-Format an Controller
                            ;ausgeben
        LD      L,A         ;Berechnung der Adresse des
        LD      H,0         ;Disk-Headerblocks,
        ADD     HL,HL       ;indem zu DHB die
        ADD     HL,HL       ;Drive-Nummer addiert
        ADD     HL,HL       ;wird
        ADD     HL,HL
        LD      DE,DHB      ;Drive-Nummer addieren
        ADD     HL,DE
        RET
;
;
;SET TRACK
;
SETTRA: IN      A,(FLOPT)   ;Track-Reg. lesen
        CP      C           ;Auf richtigem Track?
        RET     Z           ;Ja - Return
        LD      A,C         ;Akku mit gewünscht. Spur
        OUT     (FLOPD),A   ;laden und ausgeben
        LD      A,1CH+STEP  ;Kopf auf der Spur
        OUT     (FLOPC),A   ;positionieren
        CALL    WAIT
        RET
;
;
;SET SEKTOR
;
SETSEC: LD      A,C         ;Sektornummer laden
        OUT     (FLOPS),A   ;und an Sektorreg.
        RET                 ;ausgeben
;
;
;SET DMA-ADRESSE
;
SETDMA: LD      (DMAADR),BC ;Nur Adresse speichern
        RET
;
;
;READ SEKTOR
;
READ:   LD      HL,(DMAADR) ;Zieladresse holen
        LD      B,128       ;Sektorlänge in B-Reg.
        LD      C,FLOPD     ;Daten im C-Reg.
        LD      A,8CH       ;Read-Kommando laden
        OUT     (FLOPC),A   ;und ausgeben
READL:  IN      A,(FLOPC)   ;Controller Status lesen
        BIT     0,A         ;vorzeitig fertig-Fehler
        JR      Z,ERROR     ;Fehler melden
        BIT     1,A         ;Lesen beendet?
        JR      Z,READL     ;Nein-weiterlesen
```

```
        INI                     ;Daten abspeichern
        JR      NZ,READL        ;weiterlesen,bis Sektorende
        IN      A,(FLOPC)       ;Fehler aufgetreten?
        AND     01CH            ;Fehlerbit maskieren
        RET
;
;
;WRITE SEKTOR
;
WRITE:  LD      HL,(DMAADR)     ;Quelladresse holen
        LD      B,128           ;Sektorlänge in B-Reg.
        LD      C,FLOPD         ;Daten im C-Reg.
        LD      A,0ACH          ;Write-Kommando laden
        OUT     (FLOPC),A       ;und ausgeben
WRITEL: IN      A,(FLOPC)       ;Controller Status lesen
        BIT     0,A             ;vorzeitig fertig-Fehler
        JR      Z,ERROR         ;Fehler melden
        BIT     1,A             ;Schreiben beendet?
        JR      Z,WRITEL        ;Nein-weiterschreiben
        OUTI                    ;Daten ausgeben
        JR      NZ,WRITEL       ;bis Sektorende
        IN      A,(FLOPC)       ;Fehler aufgetreten?
        AND     01CH            ;Fehlerbit maskieren
        RET
;
;
;
ERROR:  LD      A,1             ;Fehler kennzeichnen
        RET
;
;
;Warten, bis das Kommando ausgeführt ist
;
WAIT:   LD      B,30            ;Verzögerungskonstante
WLOOP:  DJNZ    WLOOP           ;Warten, bis Status stabil ist
WAIT1:  IN      A,(FLOPC)       ;Status lesen
        BIT     0,A             ;BUSY gesetzt ?
        JR      NZ,WAIT1        ;Ja - weiterwarten
        RET
;
;
;Datenbereiche
;
DMAADR: DEFW    0
;
;
;Disk-Header-Block
;
DHB:    DEFW    TRANS
        DEFW    0
        DEFW    0
        DEFW    0
        DEFW    DIRBUF
        DEFW    DPB
        DEFW    CHK00
        DEFW    ALL00
;
;
;Sektor-Translation-Table
;
TRANS:  DEFB    1,7,13,19,25 ;SKEW-Faktor 6
        DEFB    5,11,17,23   ;Für 8"-Laufwerke
        DEFB    3,9,15,21
        DEFB    2,8,14,20,26
        DEFB    6,12,18,24
        DEFB    4,10,16,22
;
;
;Disk-Parameter-Block
;
DPB:    DEFW    26
        DEFB    3
        DEFB    7
        DEFB    0
        DEFW    242
        DEFW    63
        DEFB    192
        DEFB    0
        DEFW    16
        DEFW    2
;
;
;Puffer für Directory im RAM
;
DIRBUF: DEFS    128
;
;
;Disk-Allocation-Table
;
ALL00:  DEFS    31
;
;
;Check Disk
;
CHK00:  DEFS    16
;
;
        END
```

BIOS-Routinen für Z80-Systeme zur Steuerung des Floppy-Controllers

NEU: CEPAC-65 (nach c't 3/'84)

Der preisgünstige Single-Board-Computer für Festprogramm-Anwendungen



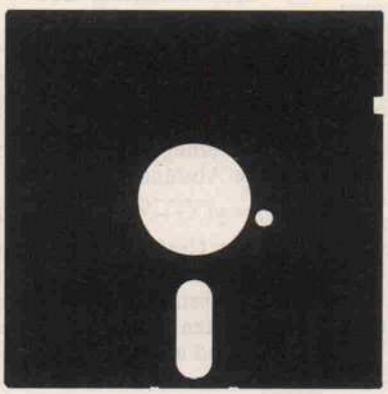
Bausätze: Version A (kleine Platine, ohne EPROM und Steckerleiste, NMOS) ..... **DM 69,—**
Version B (große Platine mit Steckerleiste, sonst wie A) ..... **DM 89,—**

Massenweise MONITORE! Massenweise

Heath-Zenith ZVM 122 (bernstein) ..... **DM 295,—**
(12 Zoll, 15 MHz) ZVM 123 (grün) ..... **DM 355,—**
PRINCE (bernstein) ..... **DM 555,—**
(12 Zoll, 25 MHz) (grün) ..... **DM 495,—**
grün nachleuchtend ..... **DM 555,—**
LOW-COST-DRUCKER
MANNESMANN-TALLY MT 80 ..... **DM 995,—**

TANDON-FLOPPY-Laufwerke
— u. a. von IBM für gut befunden ...

TM 50-1 (ss/dd, 250 KB, 1 x 40 Track, z. B. für Apple) **DM 586,—**
TM 50-2 (ds/dd, 500 KB, 2 x 40 Track, z. B. für c't 86) **DM 698,—**
TM 55-2 (ds/dd, 500 KB, 2 x 40 Track) ..... **DM 798,—**
TM 55-4 (ds/qd, 1 MB, 2 x 80 Track) ..... **DM 948,—**
TM 55-2 + 55-4 sind mikroprozessorgesteuert (extrem genaue Kopfpositionierung!)
Alle Laufwerke in Slimline-Ausführung (½ Bauhöhe).
P.S. Tandon hat einen Weltmarktanteil von 75 % !



Mit uns speichern Sie richtig!

5,25"	10 Stck.
SS/SD 48TPI	52,—
SS/DD 48TPI	58,—
SS/QD 96TPI	74,—
DS/DD 48TPI	79,50
DS/QD 96TPI	84,—
8"	
SS/SD	58,—
SS/DD	67,—
DS/DD	79,50

Größere Mengen Preis auf Anfrage. Preise pro Stück inkl. MwSt. - zzgl. Versandkostenanteil: DM 5,—.

Leerplatinen für c't-Projekte:
c't-Terminal — Version A — (ohne Tastatur) ..... **DM 59,—**
c't-Terminal — Version B — (mit Tastatur) ..... **DM 75,—**
Universelles Netzteil ..... **DM 14,—**
c't-86-System: Prozessor-Karte ..... **DM 85,—**
RAM-Karte ..... **DM 88,—**
I/O-Karte ..... **DM 69,—**
Floppy-Karte ..... **DM 65,—**
BUS-Platine (96pol.) für 10 Steckplätze ..... **DM 49,—**
c't-Sprachsynthesizer ..... **DM 21,—**
CEPAC-65, Version A (100 x 80 mm) ..... **DM 27,—**
Version B (100 x 160 mm) ..... **DM 52,—**

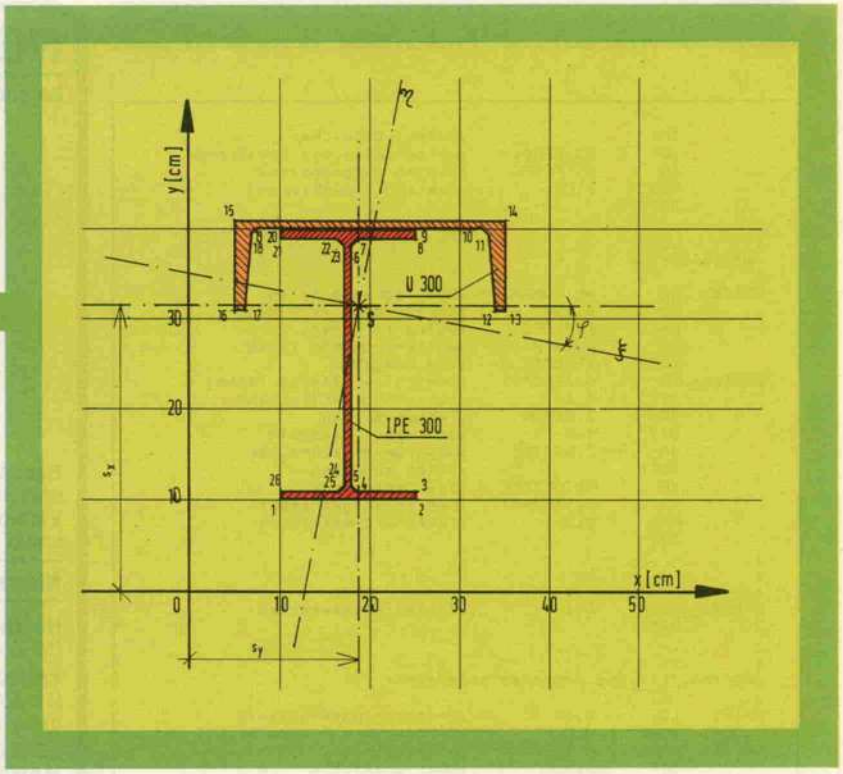
VERSAND: per NN (+ Versandkosten) oder per Vorauskasse (V-Scheck oder Überweisung auf Pschkto Han 1429 28-308, keine Versandkosten). Ausland nur gegen Vorauszahlung (+ DM 15,— Versandkosten). Alle Preise inkl. MwSt.

MARFLOW-COMPUTING GmbH
Brüderstraße 2 · 3000 Hannover 1 · Telefon 05 11/32 60 98

Polygone

Vom Umfang bis zu den Flächenmomenten

Hermann Malcharczyk



Die Ermittlung von Querschnittswerten geradlinig begrenzter Flächen, wie beispielsweise Flächeninhalte, Schwerpunktlage und sonstige Werte, ist eine Aufgabe, die man in der Technik häufig zu lösen hat. Bei einfachen Flächen wie Dreiecken oder Vierecken dürfte man die entsprechenden Formeln noch in der Erinnerung oder in einer Formelsammlung finden. Aber was tun, wenn's komplizierter wird? Nun, für 'genormte' Flächen, wie zum Beispiel für den Querschnitt von Stahlträgern, gibt es Tabellenwerke. Wenn aber alle Stricke reißen, dann müssen Sie selbst rechnen und haben ein schönes Stück Arbeit vor sich.

Die Berechnung wird bei entsprechend vielen Eckpunkten so aufwendig und damit fehleranfällig, daß man sie besser seinem Computer überläßt. Das hier vorgestellte Programm wurde auf einem ZX 81 geschrieben, läßt sich aber wegen seiner Transparenz und der Verwendung 'transportabler' BASIC-Befehle leicht für andere Systeme umstricken oder dem persönlichen Bedarf anpassen.

Die Eckpunkte des zu berechnenden Polygons werden in Form von kartesischen Koordinaten in einem System mit willkürlich gewählter Lage zu den Achsen eingegeben und vom Rechner weiterverarbeitet. Anhand der noch recht einfachen Berechnung des Umfanges und des Flächeninhaltes sollen dem Interessierten die theoretischen Grundlagen nähergebracht werden.

Berechnung des Umfanges

Der horizontale Abstand der Punkte $\overline{P_1P_2}$ in Bild 1 beträgt X_2-X_1, der vertikale Y_2-Y_1. Nach Pythagoras beträgt der direkte Abstand

$$\overline{P_1P_2} = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}.$$

Um den Umfang einer Fläche zu ermitteln, muß man die einzelnen Abstände zwischen den Eckpunkten wie nach Bild 1 erfassen und anschließend addieren.

Berechnung des Flächeninhaltes

Nun wird's allmählich komplizierter, daher sei das Verfahren anhand der einfachsten Fläche, dem Dreieck, vorgestellt. Die allgemeine Formel zur Flächenermittlung lautet:

$$A = \frac{1}{2} \cdot \sum_{1}^{n} \left((X_i - X_{i+1})\,(Y_i + Y_{i+1}) \right)$$

(siehe Programmzeile 441)

Bei einem Dreieck, also bei $n=3$, ergibt sich

$$A = \frac{1}{2}\Big((X_1-X_2)\,(Y_1+Y_2) + (X_2-X_3)\,(Y_2+Y_3) + (X_3-X_1)\,(Y_3+Y_1) \Big)$$

Stellt man die Formel um, wird sichtbar, daß bei der Berechnung der Dreiecksfläche die Fläche von 3 Trapezen ermittelt wird:

$$A = \frac{Y_1+Y_2}{2}(X_1-X_2) \quad ①$$

$$+ \frac{Y_2+Y_3}{2}(X_2-X_3) \quad ②$$

$$+ \frac{Y_3+Y_1}{2}(X_3-X_1) \quad ③$$

Die farbig abgesetzte Fläche ③ ist wegen $X_1 > X_3$ negativ, es verbleibt also genau das Dreieck $P_1P_2P_3$. Doch nun genug der Theorie, schließlich soll das Programm benutzt werden, und es kann noch erheblich mehr.

Dateneingabe

Haben Sie das Listing einmal auf Kassette geladen, startet das Programm selbständig und 'fragt Sie ab'.

- Anzahl der Eckpunkte?
- Maßeinheit?
- X- und Y-Koordinaten der Eckpunkte?
- Korrekturen falsch eingegebener Koordinaten?

Eine grobe Übersicht über das Programm gibt Ihnen das stark vereinfachte Flußdiagramm in Bild 3. Im folgenden wird auch gleich immer auf die relevanten Zeilen im Programmlisting eingegangen.

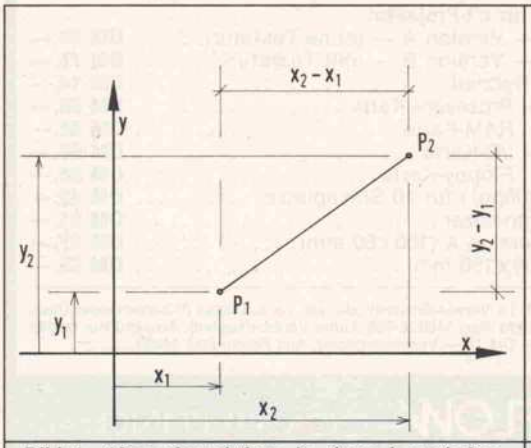


Bild 1. Abstandsermittlung in einem kartesischen Koordinatensystem

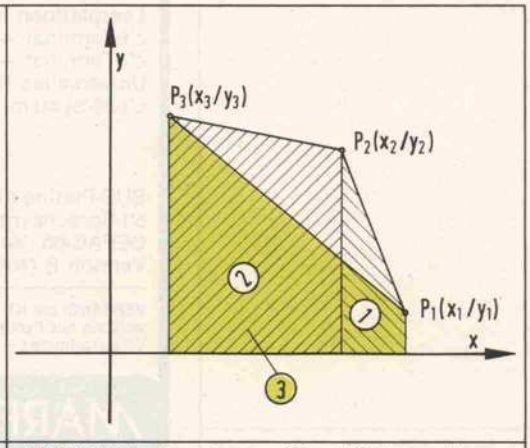
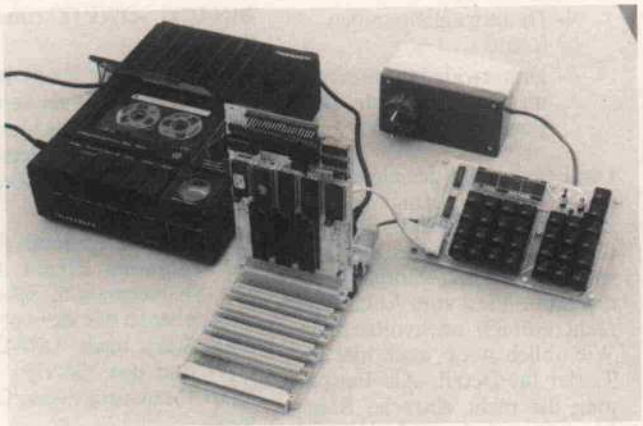


Bild 2. Flächenberechnung; ① und ② kennzeichnen die Flächen mit gleicher Schraffur

COBOLD

IHR Lern- und Proficomputer auf drei Platinen!
Der ideale Einstieg in die Microprozessortechnik



COBOLD — ein Computer mit zauberhaften Qualitäten dank eines neuen, raffinierten Hardware-Konzepts und eines sagenhaft komfortablen Betriebssystems. Auf drei Platinen.
— ein Maschinensprache-Computer auf Basis 6502/65C02, der auch Textverarbeitung, BASIC und FORTH kann.
— der sinnvollste Einstieg in die Microprozessortechnik.
— der Computer für alle — auch Ihre — Problemstellungen.
— beschrieben mit Bauanleitung in ELRAD 3, 4 + 5/83.

Lernen auch Sie zaubern wie ein Cobold — steigen Sie ein in die Microprozessortechnik mit dem neuen elrad-COBOLD-System! Fordern Sie Prospekte an!

Die Komplett-Ausstattungen:

GRUNDVERSION: (CIM 65-Prozessorkarte, Basis- und TD-Platine) mit CPU 6502. RIOT 6532, 2 K RAM, Monitor-EPROM. Basisplatine bestückt mit 1 Federleiste.
Bausatz ..... **DM 298,—**
Bausatz mit fertiger CPU-Karte ..... **DM 389,—**
Fertig aufgebautes System ..... **DM 449,—**

ERWEITERTE VERSION (Grundversion mit 4 K RAM, 3x RIOT 6532, Basisplatine mit 5 Federleisten).
Bausatz ..... **DM 398,—**
Bausatz mit fertiger CPU-Karte ..... **DM 498,—**
Fertig aufgebautes System ..... **DM 549,—**

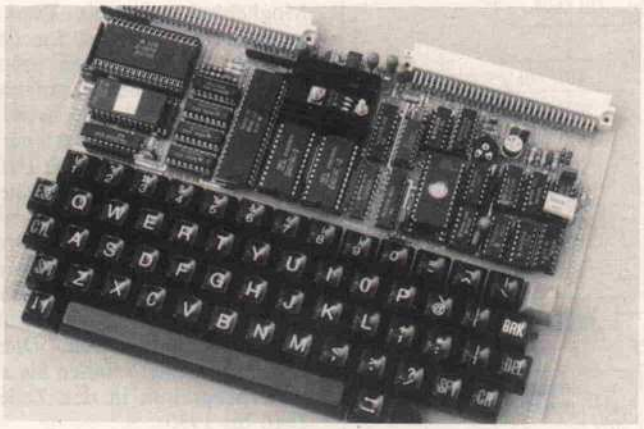
NETZTEIL für den COBOLD und das c't-Terminal im Steckergehäuse **DM 49,—** (Bausatz) bzw. **DM 86,—** (fertig)

DAS HANDBUCH für den COBOLD: „6502/65C02 Maschinensprache" . **DM 48,—**

Intelligentes Terminal
mit professionellen Attributen:

c't-Terminal

DER Terminal-Computer auf Doppel-Euro-Karte
mit oder ohne integrierter Tastatur!



— beschrieben in c't Nr. 12/83 und 1/84
— 6511-Singlechipcomputer mit 6545-Videocontroller
— 4 KB-Bildwiederholspeicher (scrollbar)
— Bildformat 80 x 24 oder 64 x 20 (per Software umschaltbar)
— Zeichenmatrix 8 x 11 (bei 80 x 24) oder 8 x 13 (bei 64 x 20)
— max. 8 Zeichensätze (inkl. Blockgrafik)
— Invers-, Blink-Modus, Breitschrift, halbe Helligkeit
— serielles Interface (V 24- oder TTL-Pegel)
— integrierte Centronics-Schnittstelle
— integrierte Spannungsregelung und -wandlung für V24
— Tastaturanschluß: 8-bit-parallel (ASCII) oder 8 x 9 Tastenmatrix
— PREISE: **Version A** (ohne Tastatur)
Bausatz **DM 449,—; DM 549,—** Fertigkarte
Platinenmaße 233 x 85 mm
Version B (mit integrierter Tastatur)
Bausatz **498,—; DM 639,—** Fertigkarte
Platinenmaße 233 x 160 mm

Prospektmaterial auf Anforderung!

Endlich:

c't-86

Das 1. <u>echte</u> 16-bit-Microcomputer-System
der Welt (auch) zum Selbstbau!

Ein Vier-Karten-System, basierend auf dem für 16 bit erweiterten ECB-Bus:
— echte 16-bit-Rechenleistung
— kein neuer, sondern ein weitverbreiteter Bus
— dadurch bereits existierende ECB-Peripherie-Karten einsetzbar
— IBM-PC-kompatibel
— Betriebssysteme CP/M-86 und MS-DOS II
— **Komplettpreis unter DM 1 700,— (Bausatz)**
— vorgestellt in Heft 1, 2 + 3/84 von c't — dem neuen Magazin für Computertechnik

Die vier Karten:

Platine 1: **CPU-KARTE mit 8086,** optional 8087 Arithmetik-Prozessor, 8259 Interrupt-Controller, 8 KB Monitorprogramm mit CP/M-86-Urlader.
Komplett-Bausatz ..... **DM 349,—; DM 449,—** Fertigkarte

Platine 2: **I/O-KARTE** mit V-24-Interface für Terminal-Anschluß, Centronics-Schnittstelle, Kassettenrekorder-Interface und Timer.
Komplett-Bausatz ..... **DM 249,—; DM 349,—** Fertigkarte

Platine 3: **FLOPPY-CONTROLLER-KARTE** zum Anschluß bis zu 4 Laufwerken 5¼ oder 8 Zoll (auch gemischt) mit dem neuen Controller-IC WD 2797.
Komplett-Bausatz ..... **DM 498,—; DM 598,—** Fertigkarte

Platine 4: **256-KB-RAM-KARTE** mit 128 oder 256 KB dyn. RAM (max. 3 Karten einsetzbar ≙ 768 KB RAM!).
Komplett-Bausatz **DM 598,—** (128 KB) bzw. **DM 899,—** (256 KB), **DM 698,—** bzw. **DM 999,—** Fertigkarte

ECB-Buskarte mit 10 Steckplätzen (96pol.) — fertig ..... **DM 169,—**

Geplante Ergänzungen: CPU-Karte mit 68000, SASI-Interface, Vollgrafikkarte, Z 80 Subprozessor-Karte.

Betriebssystem CP/M-86, angepaßt auf c't 86 ..... **DM 595,—**
CP/M-86 für IBM-PC ..... **DM 188,10**
Leerplatinen, Floppy-Laufwerke, Netzteile und Gehäuse ..... **auf Anfrage**

KOMPLETTSYSTEM, anschlußfähig im Gehäuse, wie abgebildet, mit 256 KB RAM und 2 x 500 KB-Floppy-Laufwerken ..... **DM 3 985,—**
Portable in Vorbereitung.

Fordern Sie Prospekte an!

VERSAND: per NN (+ Versandkosten) oder per Vorauskasse (V-Scheck oder Überweisung auf Pschkto Han 1429 28-308, keine Versandkosten). Ausland nur gegen Vorauszahlung (+ DM 15,— Versandkosten). Alle Preise inkl. MwSt. **Händleranfragen erwünscht.**

MARFLOW–COMPUTING GmbH

Brüderstraße 2 · 3000 Hannover 1 · Telefon 05 11/32 60 98

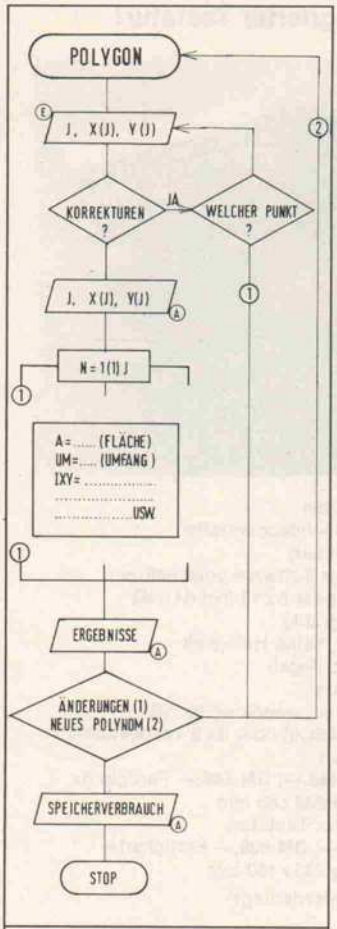


Bild 3. Grobe Übersicht über den Programmablauf

Nach Eingabe der Anzahl der Eckpunkte (J) und der von Ihnen gewünschten Maßeinheit (also cm, m, usw.) werden die Speicherplätze für die Koordinaten dieser Eckpunkte eröffnet, die in der FOR Z=1 TO J / NEXT Z — Schleife (Zeile 330—391) abgefragt und auf den Bildschirm geschrieben werden. Die PRINT-Zeile ist (S+5), wobei für Z<17 S=Z gesetzt ist und für Z≧17 SCROLL ausgeführt und S=16 gesetzt wird.

Nach erfolgter Eingabe bietet die Frage 'Korrekturen?' die Möglichkeit, einzelne Punkte zu ändern. Dieses ist für den praktischen Betrieb sehr nützlich, da zum einen Tippfehler die unangenehme Eigenschaft haben, daß man sie erst nach dem Tippen von 'NEWLINE' bemerkt, zum anderen können Sie diese Korrekturschleife auch nach Beendigung des Programmdurchlaufs benutzen, um durch Veränderung einzelner Eckpunktkoordinaten die Fläche zu optimieren. Diese Korrekturschleife finden Sie als Unterprogramm in den Zeilen 1810 bis 1930.

Verneinen Sie die Frage nach Korrekturen, so wird das Polygon dadurch geschlossen, daß X(J+1) = X(1) und Y(J+1) = Y(1) gesetzt werden. (Siehe Zeilen 410 und 420.)

Berechnungen

Folgende Berechnungen kann das Programm durchführen:

1. Bezogen auf das angenommene Koordinatensystem
   - — Abwicklung (Umfang)
   - — Fläche
   - — statische Momente M_{sx} und M_{sy}
   - — Trägheitsmomente I_x, I_y und I_{xy}
2. Bezogen auf die Schwerachsen (parallel zum angenommenen Koordinatensystem)
   - — Schwerpunktabstände s_x, s_y
   - — Trägheitsmomente I_x, I_y, I_{xy}
   - — Widerstandsmomente W_{xo}, W_{xu}, W_{yr}, W_{yl}
3. Bezogen auf die Hauptachsen
   - — Winkel zwischen ξ- und x-Achse
   - — Hauptträgheitsmomente I_ξ und I_η
   - — Hauptträgheitsradien i_ξ und i_η
   - — Hauptwiderstandsmomente für jeden Eckpunkt W_ξ und W_η

Die Berechnung des eingegebenen Polygons erfolgt in den Zeilen 440 bis 1690 und ist, da die Variablenbezeichnungen weitgehend beibehalten wurden, zumindest vom Ablauf her recht einfach nachvollziehbar. Wie üblich steckt auch hier der Teufel im Detail. Als Beispiel mag die recht einfache Beziehung für die Lage der Hauptachsen dienen:

$$\tan 2\alpha = 2\, I_{xy}/(I_y - I_x)$$

Wenn nun α als Winkel zwischen ξ- und x-Achse ausgegeben werden soll, so gibt es einige Sonderfälle vorab zu klären:

$I_{xy} = 0$ und $I_x = I_y$, $\Rightarrow \alpha = 0°$

$I_{xy} = 0$ und $I_x < I_y$, $\Rightarrow \alpha = 90°$

usw. siehe Zeilen 1140 bis 1205.

Sollten Sie nicht den gesamten Umfang des Programms nutzen wollen (wen interessieren schon die Hauptachsen oder Trägheitsmomente einer Grundstücksfläche?), so können Sie die entsprechenden Programmzeilen und Ausdruckbefehle löschen.

Ergebnisse

Im Bereich des Ergebnisausdrucks (ab Zeile 1997) wird nicht nur gedruckt, sondern auch gerechnet, da andernfalls zu viel Speicherplatz benötigt würde. Die Ergebnisse lassen trotz des für die Berechnung eingeschalteten FAST-Modus eine Weile auf sich warten, bei dem im Programmlisting vorgeführten Beispiel mit 26 Eckpunkten von der Meldung 'ICH DENKE' bis 'GLEICH BIN ICH SOWEIT' rund 2 Minuten.

Für 'printerlose' ZX81-Besitzer und Papiersparer sei vermerkt, daß sich die Pausen hinter den Ausgabeblöcken verlängern lassen, so daß man Zeit zum Abschreiben findet.

Daß der 16-K-Speicher für dieses Programm Pflicht ist, haben Sie vermutlich schon geahnt, ebenso wie der automatische Start nach LOAD''POLY'' und das 'SAVEn' durch RUN (beziehungsweise GOTO) 9990.

Praxis

Die Anzahl der Eckpunkte ist nach oben nicht begrenzt, sie muß jedoch mindestens 3 betragen. Bei der Durchnumerierung der Eckpunkte beachten Sie bitte, daß Sie die zu berechnende Fläche in mathematisch positiver Richtung entsprechend den nachfolgenden Beispielen durchnumerieren, also entgegen dem Uhrzeigersinn bei positiven Flächen. Stellen Sie sich am besten vor, Sie würden die Fläche umfahren und müßten sie stets zu Ihrer Linken behalten. So lassen sich auch Fehlflächen (Löcher) berücksichtigen (Bild 4). Es ist übrigens nicht erforderlich, daß Sie sich nur im 1. Quadranten befinden. Überschneidungen von Begrenzungslinien ohne Anordnung von Eckpunkten sind dagegen nicht zulässig, und im übrigen mit dem oben Gesagten nicht vereinbar; sie lassen sich jedoch stets vermeiden (siehe Bild 5).

Bei nicht geradlinig begrenzten Flächen erhalten Sie für die Praxis wohl immer hinreichend genaue Ergebnisse, wenn Sie hier mit einem Polygonzug die

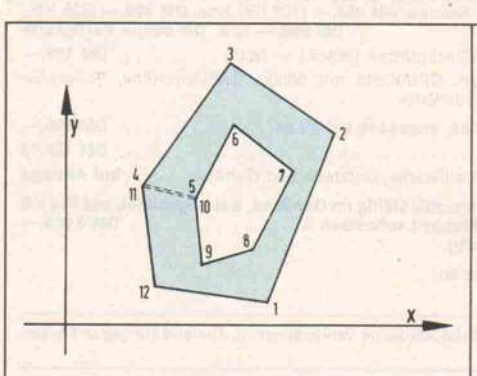


Bild 4. Leerflächen lassen sich berücksichtigen

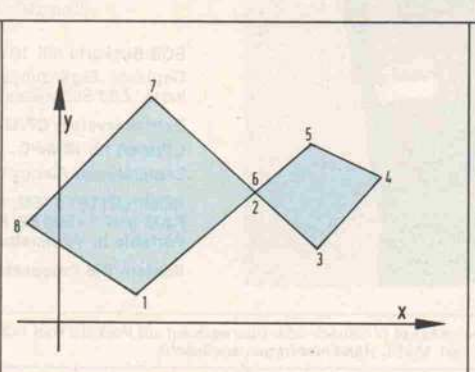


Bild 5. Die Flächen müssen nicht nur im 1. Quadranten liegen

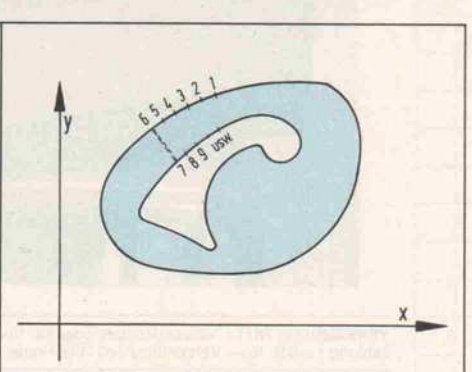


Bild 6. Auch gekrümmte Linien lassen sich annähern

```
  5 REM "POLY"
  6 PRINT "
 10 PRINT "  BERECHNUNG VON
 15 PRINT "  QUERSCHNITTSWERTEN
 20 PRINT "  EINES POLYGONS MIT
N ECKEN    "
 30 PRINT "
 40 PRINT
 45 PRINT
 46 PRINT
 47 PRINT
 48 PRINT
 49 PRINT
 50 PRINT "PROGRAMM VON"
 55 PRINT
 60 PRINT "HERMANN MALCHARCZYK"
 70 PRINT "DIPL.-ING. FUER DAS
BAUWESEN"
 80 PRINT "LOTHRINGER STRASSE 5
"
 90 PRINT "2870 DELMENHORST"
100 PRINT "TEL.:04221/22647"
104 PAUSE 50
105 COPY
120 CLS
192 PRINT "ANZAHL DER ECKEN :"
210 INPUT J
220 PRINT "MASSEINHEIT ."
230 INPUT E$
250 CLS
260 PRINT "POLYGON MIT ";J;" EC
KEN"
270 PRINT "----------------------
------------"
280 PRINT "  ECKE       KOORDINA
TEN"
290 PRINT
300 PRINT AT 4,11;"X (";E$;")";
AT 4,20;"Y (";E$;")"
310 DIM X(J+1)
320 DIM Y(J+1)
330 FOR Z=1 TO J
340 IF Z>=17 THEN SCROLL
345 LET S=Z
346 IF Z>=17 THEN LET S=16
350 PRINT AT (S+5),3;Z
360 INPUT X(Z)
370 PRINT AT (S+5),11;X(Z)
380 INPUT Y(Z)
390 PRINT AT (S+5),20,Y(Z)
391 NEXT Z
400 GOSUB 1700
402 CLS
403 PRINT AT 21,0;"ICH DENKE"
404 PAUSE 150
405 FAST
410 LET X(J+1)=X(1)
420 LET Y(J+1)=Y(1)
430 LET A=0
431 LET UM=0
432 LET SX=0
433 LET IXY=0
440 FOR N=1 TO J
441 LET A=A+.5*(X(N)-X(N+1))*(Y
(N)+Y(N+1))
445 LET DX=(X(N+1)-X(N))*(X(N+1
)-X(N))
446 LET DY=(Y(N+1)-Y(N))*(Y(N+1
)-Y(N))
447 LET UM=UM+SQR (DX+DY)
452 LET IXY=IXY+((X(N)*Y(N+1)-
X(N+1)*Y(N))*((X(N)+X(N+1))*(Y(N
)+Y(N+1))-.5*(X(N)*Y(N+1)+X(N+1)
*Y(N)))/12)
453 LET SX=SX+((X(N)*Y(N+1)-X(N
+1)*Y(N))*(Y(N)+Y(N+1)))/6
460 NEXT N
500 LET SSX=SX/A
510 LET SY=0
520 FOR O=1 TO J
530 LET SY=SY+((X(O)*Y(O+1)-X(O
+1)*Y(O))*(X(O)+X(O+1)))/6
540 NEXT O
550 LET SSY=SY/A
560 LET IX=0
570 LET IY=0
580 FOR P=1 TO J
590 LET IX=IX+((X(P)*Y(P+1)-X(P
+1)*Y(P))*((X(P)*X(P)+2*X(P)*X(
P+1)+Y(P+1)*Y(P+1)))-Y(P)*Y(P+1)
))/12
600 LET IY=IY+((X(P)*Y(P+1)-X(P
+1)*Y(P))*((X(P)*X(P)+2*X(P)*X(
P+1)+X(P+1)*X(P+1)))-X(P)*X(P+1)
))/12
610 NEXT P
1000 DIM Q(J+1)
1010 DIM R(J+1)
1030 FOR S=1 TO (J+1)
1040 FOR T=1 TO (J+1)
1050 LET Q(S)=X(S)-SSY
1060 LET R(S)=Y(S)-SSX
1070 NEXT T
1080 NEXT S
1090 LET IXS=0
1091 LET IXYS=0
1095 LET IYS=0
1099 FOR U=1 TO J
1110 LET IXS=IXS+((Q(U)*R(U+1)-Q
(U+1)*R(U))*(((R(U)*R(U)+2*R(U)*
R(U+1)+R(U+1)*R(U+1)))-R(U)*R(U+
1)))/12
1111 LET IXYS=IXYS+(((Q(U)*R(U+1
)-Q(U+1)*R(U))*((Q(U)+Q(U+1))*(R
(U)+R(U+1))-.5*(Q(U)*R(U+1)+Q(U+
1)*R(U))))/12)
1120 LET IYS=IYS+((Q(U)*R(U+1)-Q
(U+1)*R(U))*(((Q(U)*Q(U)+2*Q(U)*
Q(U+1)+Q(U+1)*Q(U+1)))-Q(U)*Q(U+
1)))/12
1130 NEXT U
1140 IF ABS IXYS<.00000001 AND I
XS=IYS THEN LET ALP=0
1141 IF ABS IXYS<.00000001 AND I
XS<IYS THEN LET ALP=90
1145 IF ABS IXYS<.00000001 THEN
GOTO 1210
1150 IF ABS (IXS-IYS)<.000001 AN
D ABS IXYS>.0000001 THEN LET ALP
=SGN (-IXYS)*45
1160 IF ABS (IXS-IYS)<.000001 AN
D ABS IXYS>.0000001 THEN GOTO 12
10
1200 IF IXS<=IYS THEN GOTO 1204
1201 LET TGA=2*ABS IXYS/(IXS-IYS
)
1202 LET ALP=90/PI*ATN TGA*SGN (
-IXYS)
1203 IF IXS>IYS THEN GOTO 1210
1204 LET TGA=2*ABS IXYS/(IYS-IXS
)
1205 LET ALP=90/PI*ATN TGA-90
1211 LET H1=.5*(IXS+IYS)
1212 LET H2=.5*(IXS-IYS)*COS ((2
*ALP)*PI/180)
1213 LET H3=IXYS*SIN (2*ALP*PI/1
80)
1220 LET IXI=H1+H2-H3
1230 LET IET=H1-H2+H3
1240 LET GRX=X(1)
1250 LET KLX=X(1)
1260 LET GRY=Y(1)
1270 LET KLY=Y(1)
1271 LET CP=COS (ALP*PI/180)
1272 LET SP=SIN (ALP*PI/180)
1280 FOR E=1 TO J
1290 IF X(E)>GRX THEN LET GRX=X(
E)
1300 IF X(E)<KLX THEN LET KLX=X(
E)
1310 IF Y(E)>GRY THEN LET GRY=Y(
E)
1320 IF Y(E)<KLY THEN LET KLY=Y(
E)
1330 NEXT E
1340 LET WXO=IXS/(GRY-SSX)
1350 LET WXU=IXS/(SSX-KLY)
1360 LET WYR=IYS/(GRX-SSY)
1370 LET WYL=IYS/(SSY-KLX)
1400 DIM G(J)
1410 DIM H(J)
1420 FOR F=1 TO J
1430 FOR K=1 TO J
1440 LET G(F)=Q(F)*CP+R(F)*SP
1450 LET H(F)=R(F)*CP-Q(F)*SP
1460 NEXT K
1470 NEXT F
1690 GOTO 1990
1700 CLS
1710 FOR B=1 TO ((J-1)/15)+1
1715 LET KO=1
1716 CLS
1720 IF B=1 THEN PRINT "POLYGON
MIT ";J;" ECKEN"
1730 IF B=1 THEN PRINT "KOORDINA
TEN DER ECKPUNKTE"
1740 IF B=1 THEN PRINT "--------
------------------------"
1750 IF B=1 THEN PRINT "ECKE
  X(";E$;")";TAB 19;"Y(";E$;")"
1760 IF B>1 THEN PRINT AT 2,0;"F
ORTSETZUNG"
1770 FOR C=(B*15-14) TO J
1775 IF C>B*15 THEN GOTO 1801
1780 LET ZEILE=C+20-B*15
1790 PRINT AT ZEILE,3;C;AT ZEILE
,11;X(C);AT ZEILE,20;Y(C)
1800 NEXT C
1803 PAUSE 300
1804 CLS
1810 PRINT "KORREKTUREN ?"
1820 PRINT "JA=1 , NEIN=0"
1830 INPUT KO
1840 IF KO=0 THEN GOTO 1920
1845 CLS
1850 PRINT "WELCHER PUNKT ?"
1860 INPUT D
1865 CLS
1870 PRINT "RICHTIGE X-KOORDINAT
E ?"
1880 INPUT X(D)
1885 CLS
1885 PRINT "RICHTIGE Y-KOORDINAT
E ?"
1890 INPUT Y(D)
1900 GOTO 1715
1920 NEXT B
1930 CLS
1940 LPRINT "POLYGON MIT ";J;" E
CKEN"
1941 LPRINT
1945 LPRINT "KOORDINATEN DER ECK
PUNKTE"
1950 LPRINT "--------------------
------------"
1955 LPRINT "ECKE      X(";E$;")
";TAB 19;"Y(";E$;")"
1956 LPRINT
1960 FOR D=1 TO J
1965 LPRINT TAB 3;D;TAB 11;X(D);
TAB 20;Y(D)
1970 NEXT D
1971 LPRINT
1972 LPRINT
1973 LPRINT
1974 LPRINT
1980 RETURN
1990 IF ABS IXYS<=.0000001 THEN
LET IXYS=0
1996 CLS
1997 PRINT "GLEICH BIN ICH SOWEI
T"
1998 PAUSE 150
1999 CLS
2000 SLOW
2009 PRINT "ERGEBNISSE"
2010 PRINT
2011 PRINT "STATISCHE WERTE BEZO
GEN"
2012 PRINT "AUF DAS ANGENOMMENE
SYSTEM"
2013 PRINT "--------------------
------------"
2014 PRINT "ABWICKLUNG"
2015 PRINT "  U    = ";UM;" ";E$
2016 PRINT
2020 PRINT "FLAECHE"
2030 PRINT "  A    = ";A;" ";E$;"
**2"
2031 PRINT
2032 PRINT "STATISCHE MOMENTE"
2033 PRINT "  MSX = ";SX;" ";E$;
"**3"
2034 PRINT "  MSY = ";SY;" ";E$;
"**3"
2065 PRINT
2070 PRINT "TRAEGHEITSMOMENTE"
2080 PRINT "  IX   = ";IX;" ";E$;
"**4"
2090 PRINT "  IY   = ";IY;" ";E$;
"**4"
2094 PRINT "  IXY  = ";IXY;" ";E$
;"**4"
2095 PRINT
2096 PAUSE 100
2097 COPY
2290 CLS
2300 PRINT
2310 PRINT
2320 PRINT "STATISCHE WERTE BEZO
GEN"
2330 PRINT "AUF DIE SCHWERACHSEN
"
2340 PRINT "--------------------
------------"
2341 IF ABS SSX<.00001 THEN LET
SSX=0
2342 IF ABS SSY<.00001 THEN LET
SSY=0
2349 PRINT "SCHWERPUNKTLAGE"
2350 PRINT "  SX   = ";SSX;" ";E$
2360 PRINT "  SY   = ";SSY;" ";E$
2370 PRINT
2380 PRINT "TRAEGHEITSMOMENTE"
2390 PRINT "  IX   = ";IXS;" ";E$
;"**4"
2400 PRINT "  IY   = ";IYS;" ";E$
;"**4"
2410 PRINT "  IXY  = ";IXYS;" ";E
$;"**4"
2420 PRINT
2430 PRINT "WIDERSTANDSMOMENTE"
2450 PRINT "  WXO  = ";WXO;" ";E$
;"**3"
2460 PRINT "  WXU  = ";WXU;" ";E$
;"**3"
2470 PRINT "  WYR  = ";WYR;" ";E$
;"**3"
2480 PRINT "  WYL  = ";WYL;" ";E$
;"**3"
2485 PAUSE 100
2490 COPY
2510 CLS
2520 PRINT
2525 PRINT
2530 PRINT "STATISCHE WERTE BEZO
GEN AUF"
2540 PRINT "DIE HAUPTACHSEN"
2550 PRINT "--------------------
------------"
2560 PRINT "WINKEL ZWISCHEN XI-
UND X-ACHSE"
2565 IF ABS ALP<.0001 THEN LET A
LP=0
2570 PRINT "  = ";ALP;" GRAD"
2575 PRINT
2580 PRINT "TRAEGHEITSMOMENTE"
2590 PRINT "  I XI  = ";IXI;" ";
E$;"**4"
2600 PRINT "  I ETA = ";IET;" ";
E$;"**4"
2605 PRINT
2610 PRINT "TRAEGHEITSRADIEN"
2620 PRINT "  I XI  = ";SQR (IXI
/A);" ";E$
2630 PRINT "  I ETA = ";SQR (IET
/A);" ";E$
2640 PAUSE 100
2690 COPY
2710 CLS
2720 FOR R=1 TO ((J-1)/15+1)
2730 IF R=1 THEN PRINT "WIDERSTA
NDSMOMENTE"
2740 IF R=1 THEN PRINT "BEZOGEN
AUF DIE HAUPTACHSEN"
2750 IF R=1 THEN PRINT "--------
------------------------"
2760 IF R=1 THEN PRINT "ECKE";TA
B 8;"WXI(";E$;"**3)";TAB 20;"WE
TA(";E$;"**3)"
2770 IF R>1 THEN PRINT AT 2,0;"F
ORTSETZUNG"
2780 FOR L=(R*15-14) TO J
2790 IF L>R*15 THEN GOTO 2830
2800 LET ZEI=L+20-R*15
2801 IF ABS H(L)<.00001 THEN PRI
NT AT ZEI,3;L;AT ZEI,9;"--";AT Z
EI,21;ABS (IET/G(L))
2802 IF ABS G(L)<.00001 THEN PRI
NT AT ZEI,3;L;AT ZEI,9;ABS (IXI/
H(L));AT ZEI,21;"--"
2803 IF ABS H(L)<.00001 AND ABS
G(L)<.00001 THEN PRINT AT ZEI,3;
L;AT ZEI,9;"--";AT ZEI,21;"---"
2804 IF ABS H(L)<.00001 OR ABS G
(L)<.00001 THEN GOTO 2820
2810 PRINT AT ZEI,3;L;AT ZEI,9;A
BS (IXI/H(L));AT ZEI,21;ABS (IET
/G(L))
2820 NEXT L
2825 PAUSE 100
2830 COPY
2850 CLS
2860 NEXT R
2890 CLS
3000 PRINT "- ENDE -"
3010 PRINT
3011 PRINT
3012 PRINT
3100 PRINT "1.) WOLLEN SIE DIE K
OORDINATEN"
3110 PRINT "     VERAENDERN ?"
3111 PRINT
3120 PRINT "2.) WOLLEN SIE EIN N
EUES POLY-"
3130 PRINT "     GON BERECHNEN ?"
```

Das Polygonberechnungsprogramm POLY für den ZX81

Eckpunkte entsprechend dicht aneinander legen (Bild 6).

Sie geben alle Eckpunkte mit ihren Koordinaten der Reihe nach ein und erhalten sie in Gruppen zu 15 Stück nochmal auf dem Bildschirm zur Kontrolle vorgeführt. Sie können nun eventuelle Tippfehler korrigieren, bis Sie die Frage 'Korrekturen, Ja = 1, Nein = Ø' mit '0' beantworten.

Danach erhalten Sie die nächsten 15 Eckpunkte. Schließlich werden die Eckpunkte mit ihren richtigen Koordinaten ausgedruckt.

Nun läuft das eigentliche Rechenprogramm ab. Dabei sind vielleicht noch die Zeilen 1030 bis 1070 interessant, wo das Polygon durch Parallelverschiebung in die Schwerachsen verschoben wird, und die Zeilen 1420 bis 1470, wo das Polygon in die Hauptachsen hineingedreht wird.

Nach Abschluß des Programms haben Sie die Möglichkeit zur Veränderung einzelner Koordinatenpunkte, so daß sich Polygon-Variationen oder Querschnittsoptimierungen schnell durchführen lassen.

Anwendungsgebiete

Das hier vorgestellte Polygonprogramm erweitert den sonst üblichen Anwendungsbereich des ZX81 in eine neue Richtung. Wie die nachfolgend abgebildeten möglichen Anwendungen zeigen, lassen sich mit Hilfe dieses Programms zeitaufwendige und fehleranfällige Berechnungen schnell und sicher durchführen.

So stellt Bild 7 eine typische Problemstellung aus dem Gebiet der Vermessungstechnik dar, Bild 8 wurde aus dem Bereich des Spann-Betonbaus gewählt.

Der Leser wird sicherlich seinen persönlichen Anwendungsbereich schon abgesteckt haben, gleichgültig, ob das Programm ihm 'nur' zur Berechnung von Flächeninhalten dient oder als Anregung, es zu erweitern, um zum Beispiel von dreidimensionalen Körpern Volumen und Oberflächen berechnen zu können. □

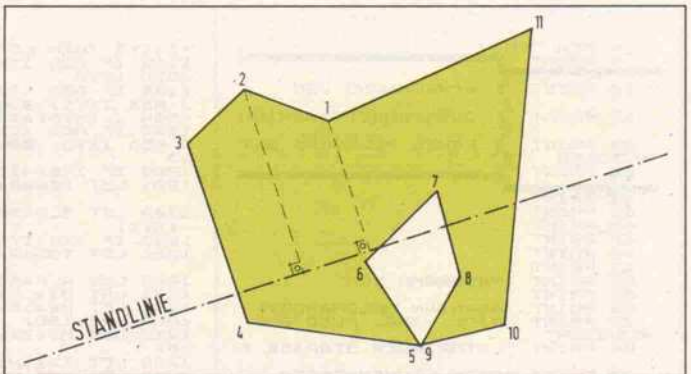


Bild 7. Die Gesamtfläche wird durch Linien zu den Eckpunkten senkrecht zur Standlinie unterteilt

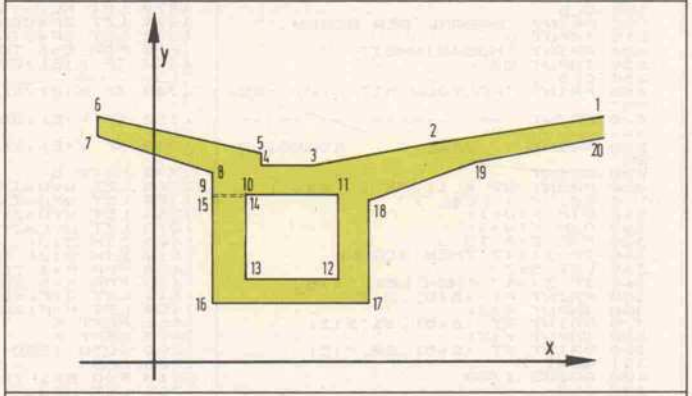


Bild 8. Spannungsermittlung an einem Brückenprofil

```
3131 PRINT
3140 PRINT "3.) WOLLEN SIE KEINE
 POLYGONE "
3150 PRINT "     MEHR BERECHNEN ?
"
3151 PRINT
3160 PRINT
3170 PRINT "BITTE WAEHLEN SIE;"
3180 PRINT "1,2 ODER 3"
3190 INPUT WO
3195 CLS
3200 IF WO=1 THEN GOTO 400
3210 IF WO=2 THEN GOTO 1
3211 CLS
3212 PRINT "VERBRAUCHTER SPEICHE
RPLATZ :  ";PEEK 16404+256
*PEEK 16405-16383;" BYTES"
3213 PRINT
3214 PRINT "DAVON FUER DAS PROGR
AMM :  ";PEEK 16396+256*
PEEK 16397-16509;" BYTES"
3215 PRINT "NOCH FREIER SPEICHER
PLATZ :";PEEK 16386-PEEK 16412+2
56*(PEEK 16387-PEEK 16413);" BYT
ES"
3230 PRINT AT 16,0;"ICH HOFFE IC
H HABE IHNEN "
3235 PRINT "ETWAS GEHOLFEN"
3236 PRINT
3240 PRINT "BITTE SCHALTEN SIE M
ICH AUS "
3250 PRINT
3260 PRINT "TSCHUESS - IHR ZX81"
9000 GOTO 9999
9990 SAVE "POL"
9991 GOTO 1
9999 STOP
```

```
 BERECHNUNG VON
 QUERSCHNITTSWERTEN
 EINES POLYGONS MIT N ECKEN
```

```
PROGRAMM VON

HERMANN MALCHARCZYK
DIPL.-ING. FUER DAS BAUWESEN
LOTHRINGER STRASSE 5
2870 DELMENHORST
TEL.: 04221/22647
```

```
POLYGON MIT 26 ECKEN
KOORDINATEN DER ECKPUNKTE
--------------------------------
ECKE        X(CM)      Y(CM)

   1          10         10
   2          25         10
   3          25         11.07
   4        18.86        11.07
   5        17.86        12.07
   6        17.86        37.93
   7        18.86        38.93
   8          25         38.93
   9          25         40
  10        32.2         40
  11        33.15        39
  12        33.8         31
  13          35         31
  14          35         41
  15           5         41
  16           5         31
  17         6.2         31
  18        6.84         33
  19        7.85         40
  20          10         40
  21          10         38.93
  22        16.15        38.93
  23        17.15        37.93
  24        17.15        12.07
  25        16.15        11.07
  26          10         11.07

ERGEBNISSE

STATISCHE WERTE BEZOGEN
AUF DAS ANGENOMMENE SYSTEM
--------------------------------
ABWICKLUNG
  U    = 181.84549 CM

FLAECHE
  A    = 112.8656 CM**2

STATISCHE MOMENTE
  MSX  = 3604.4733 CM**3
  MSY  = 2122.4068 CM**3

TRAEGHEITSMOMENTE
  IX   = 128945.71 CM**4
  IY   = 48761.694 CM**4
  IXY  = 68712.254 CM**4


STATISCHE WERTE BEZOGEN
AUF DIE SCHWERACHSEN
--------------------------------
SCHWERPUNKTLAGE
  SX   = 31.935978 CM
  SY   = 18.804727 CM

TRAEGHEITSMOMENTE
  IX   = 13833.328 CM**4
  IY   = 8850.4126 CM**4
  IXY  = 931.1162 CM**4
```

```
WIDERSTANDSMOMENTE
  WXO = 1526.1799 CM**3
  WXU = 630.62279 CM**3
  WYR = 546.48123 CM**3
  WYL = 641.11463 CM**3


STATISCHE WERTE BEZOGEN AUF
DIE HAUPTACHSEN
--------------------------------
WINKEL ZWISCHEN XI- UND X-ACHSE
  = -10.245948 GRAD

TRAEGHEITSMOMENTE
  I XI  = 14001.633 CM**4
  I ETA = 8682.1073 CM**4

TRAEGHEITSRADIEN
  I XI  = 11.138034 CM
  I ETA = 8.7706499 CM


WIDERSTANDSMOMENTE
BEZOGEN AUF DIE HAUPTACHSEN
--------------------------------
ECKE     W-XI(CM**3)  W-ETA(CM**3)

   1        604.762      1823.0218
   2        683.53353     868.35681
   3        720.5727      885.20737
   4        682.22766    2305.4502
   5        710.12195    3334.1705
   6       2443.3977     4350.0969
   7       2031.4828     2297.3554
   8       1753.6088     1789.2309
   9       1549.298      1862.2743
  10       1356.9976      739.07367
  11       1473.1113      674.60839
  12       5018.2699      581.80799
  13       7144.9116      539.14421
  14       1186.5592      606.09104
  15       2166.1012      571.31002

FORTSETZUNG

  16       4146.723       647.04442
  17       4426.547       709.48265
  18       2902.9995      666.29453
  19       2338.7247      710.6083
  20       2196.3017      859.72523
  21       2633.6886      876.23927
  22       2184.2456     2251.3227
  23       2498.4606     3222.1349
  24        705.6025     4556.8181
  25        666.57172    7899.1365
  26        633.57623    1752.9672
```

Terminal 'nach Maß'

Teil 3: Korrekturen und Hinweise für die Programmanpassung

Christian Persson

In diesem Beitrag sollen die besonders häufig gestellten Fragen zum Thema 'Anpassung des Terminal-Betriebsprogramms' beantwortet werden. Auch haben sich unterdessen — zu unserem Bedauern — sowohl in der Hardware als auch im Programm ein paar 'Makken' gefunden, die einer Korrektur bedürfen.

Empfängt das Terminal wiederholt Steuerbefehle oder Escape-Codes, die nicht vom Betriebsprogramm ausgewertet werden, so kommt es zu einem Überlaufen des Prozessor-Stack. Grund: Bei der Rückkehr aus dem Unterprogramm EXEC (vergl. c't 1/84, Seite 92) wird in diesen Fällen ein 'RTS'-Befehl übersprungen. Abhilfe: Änderung derjenigen Adressen in den Sprungtabellen, die bisher auf den Rückkehrpunkt im Hauptprogramm ($9959) weisen, in $9919. (Der Rücksprung erfolgt dann über den RTS-Befehl an der Adresse $991A).

Beim Scrollen wird die neue Zeile nur rechts vom Cursor ausgelöscht. Dadurch wird gelegentlich alter Text von unten ins Bild geschoben, wenn der Cursor rechts im Bild steht und das Scrollen mit 'Line Feed' ausgelöst wird. Abhilfe: Eine kleine Ergänzung zur Lösch-Routine, die in dem noch freien Speicherbereich zwischen $9499 und $9600 leicht untergebracht werden kann, sowie eine entsprechende Änderung des Unterprogramm-Aufrufs (siehe Tabelle 1).

Gelegentlich ist zu beobachten, daß die Bildschirmzeile vor einem mit Attribut versehenen Zeichen unerwünschterweise mit demselben Attribut abgebildet wird. Grund: Der Videocontroller adressiert während des horizontalen Strahlrücklaufs den jeweils folgenden Bereich des Bildspeichers. Abhilfe: Leider nur mit einem Eingriff in die Hardware, der das Attribut-Bit 0 (siehe c't 12/83, Seite 45) seiner bisherigen Funktion beraubt. Die Pins 9 und 10 von IC23 (74LS00) müssen von der bestehenden Verbindung abgetrennt und mit den Pins 3 und 6 von IC19 (74LS175) verbunden werden. Alle Attribute werden dann grundsätzlich während des Strahlrücklaufs, also am Ende jeder Zeile, gelöscht.

Bei der Terminal-Version B (mit Low-Cost-Tastatur) funktionierte manchmal die Shift-Taste nicht, wenn eine weitere Taste in derselben Matrix-Spalte betätigt wurde. Grund: Zwei Ausgänge von IC8 (74LS42) arbeiten dann 'gegeneinander' — in einigen Fällen (je nach Fabrikat) gewinnt der Ausgang mit L-Pegel nicht die Überhand. Dieses Problem läßt sich aus der Welt schaffen, indem man einen 74LS145 (BCD-Dekoder mit offenem Kollektor) für IC8 einsetzt. Pin 11 müßte dann theoretisch mit einem Pull-Up-Widerstand (3k3) versehen werden — es funktioniert aber auch ohne.

Die erwähnten Software-Fehler sind in der Programm-Version 1.2 bereits behoben. Eine weitere Überarbeitung des Betriebsprogramms ist zur Zeit im Gang. In diese sollen besonders häufig geäußerte Änderungswünsche und Verbesserungsvorschläge von Lesern eingehen. Sobald die Version 1.3 fertig ist, sollen die Änderungen in c't beschrieben werden. Ein 'updating' der bis dahin vom Software-Service gelieferten EPROMs wird dann gegen einen geringen Kostenbeitrag möglich sein.

```
98E5: 20 EB 95          JSR  CLRLNB     Ganze Zeile löschen

95EB: 38        CLRLNB  SEC
95EC: A5 4C             LDA  CURL       Zeiger auf Zeilenanfang
95EE: E5 4F             SBC  CROW       berechnen
95F0: 85 5A             STA  POINTL
95F2: A5 4D             LDA  CURH
95F4: E9 00             SBC  #0
95F6: 09 10             ORA  #$10       Maske für Bildspeicher
95F8: 85 5B             STA  POINTH
95FA: A2 00             LDX  #0         Spaltenzähler
95FC: 4C 57 98          JMP  CLRLNC     Zeile ab Zeiger löschen
```

Tabelle 1. Neue Zeilen vollständig gelöscht

Andere Tasten

Dem Anschluß einer beliebigen Tastatur mit Parallelausgang und Strobe steht (auch in der Terminal-Version B) kein Hindernis entgegen. Allerdings sind Tastaturen mit eigenem Encoder normalerweise teurer als die ohne Elektronik. Tastaturen mit offener Matrix lassen sich nur dann ohne Änderung der Software an dem Terminal betreiben, wenn die Belegung exakt der unserer Low-Cost-Tastatur entspricht (siehe c't 12/83, Seiten 42 und 46). Wie man dem Terminal-Prozessor 6511 die Abfrage einer anderen Tastenmatrix bis zum Format 8 x 9 übertragen kann, zeigt das folgende Beispiel.

Die Cherry-Tastatur G 80-0387 (siehe Foto) ist zum Austausch für die Folientastatur des Atari 400 entwickelt worden. Sie besitzt leichtgängige, sicher schaltende mechanische Tasten und kostet rund 125 Mark (Bezugsquelle: Fa. Raffel, Lochnerstraße 1, 4030 Ratingen 1). Die

8 x 9-Matrix ist leider nicht vollständig belegt (siehe Bild 1). Allerdings kann man die Tasten SEL und START noch zusätzlich 'einbauen', indem man die bestehenden Verbindungen auftrennt und dafür die gewünschten Reihen- und Spaltenleitungen anschließt. Die Tastenkappen lassen sich beliebig vertauschen; so sollte die Kappe mit der Beschriftung 'OPT' gegen die mit 'BREAK' ausgetauscht werden, denn sowohl die BREAK-Taste als auch die RESET-Taste schalten beim Terminal 'nach Masse'.

Alle Reihen- und Spaltenleitungen sind auf ein Flachbandkabel geführt, dessen Belegung in Bild 1 innen bezeichnet ist. Die außen angebrachten Zahlen kennzeichnen die notwendigen Verbindungen zu den Anschlußleisten X3A und X4A (c't 12/83, Seite 46). Anschluß 1 befindet sich bei beiden Leisten auf der Tastaturseite der Platine.

```
965F: A5 02     SCANR   LDA PC           PC0...3 geben Reihen-
9661: 29 F0             AND #%11110000  nummer an
9663: 05 5F             ORA CHAR        Nächsten Wert des
9665: 85 02             STA PC          Reihenzählers ausgeben
9667: A5 03             LDA PD          Spalten abfragen
9669: C9 FF             CMP #$FF        Taste gedrückt?
966B: F0 1A             BEQ NXTROW      Nein: Nächste Reihe
966D: A6 5F             LDX CHAR
966F: E0 08             CPX #$08        Reihe 8?
9671: D0 24             BNE KEYVAL      Nein: Nummer berechnen
9673: DF 03 06          BBS #4 PD SHFT  Control-Taste?
9676: F7 40             SMB #7 KMODE    Ja: Control-Flag
9678: C9 EF             CMP #%11101111  Noch eine Taste?
967A: F0 0B             BEQ NXTROW      Nein: Nächste Reihe
967C: FF 03 18 SHFT     BBS #7 PD KEYVAL Shift-Taste?
967F: E7 40             SMB #6 KMODE    Ja: Shift-Flag
9681: 57 40             RMB #5 KMODE    Shift-Lock löschen
9683: C9 7F             CMP #%01111111  Noch eine?
9685: D0 10             BNE KEYVAL      Ja: Nummer berechnen
9687: C6 5F     NXTROW  DEC CHAR        Nächste Reihe
9689: 10 95             BPL BFTEST      Vorher Puffer abfragen
```

Tabelle 2. Tastaturabfrage für 9 x 8-Matrix

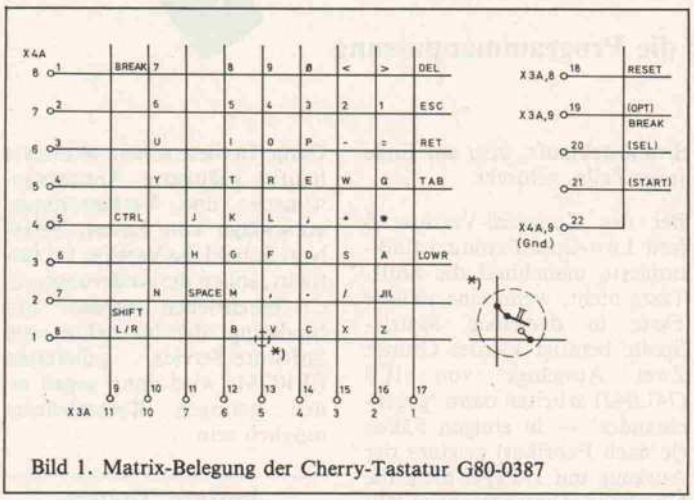


Bild 1. Matrix-Belegung der Cherry-Tastatur G80-0387

Tabelle 2 zeigt den für die Tastaturabfrage zuständigen Teil der INPUT-Routine mit den erforderlichen Änderungen. Der Prozessor gibt einen Vier-Bit-Code an den BCD-Dekoder IC8, den dieser in einen 1-aus-10-Code umsetzt. Damit wird jeweils eine Reihenleitung (beginnend mit der Reihe 8 — X3A, 11) auf log. 0 gezogen. Betätigt man eine Taste, so werden Reihen- und Spaltenleitung verbunden, und die '0' wird 'weitergereicht' an den Port D des Prozessors, mit welchem die acht Spaltenleitungen verbunden sind. Alle Port-D-Eingänge liegen sonst durch Pull-Up-Widerstände auf log. 1. Liest der Prozessor an Port D einen Hex-Wert ungleich FF, so erkennt er die Tastenbetätigung. Aus den 'Koordinaten' (Reihe, Spalte) ermittelt er später eine Tastennummer. (Diese ergibt sich aus der Spaltennummer — 0 bis 7 — zuzüglich 8 je Reihe. Die Taste 'A' hat demnach die vorläufige Nummer $0D).

'Shift' und 'Control' werden zuvor 'ausgesondert': Bei Betätigen einer Taste in der Reihe 8 prüft der Prozessor Bit 4 und Bit 7 des an Port D anliegenden Datenworts. Wenn dieses den Wert 0 hat, ist die Control- beziehungsweise Shift-Taste gedrückt. Es wird dann das entsprechende Flag-Bit in der Speicherzelle KMODE ($40) gesetzt; die Matrixabfrage wird fortgesetzt. Das Betätigen der Shift-Taste bewirkt, wie bei einer Schreibmaschine, gleichzeitig die Rückkehr aus dem 'Shift-Lock'-Modus. Das Programm setzt voraus, daß Shift und Control in der zuerst abgefragten Tastenreihe 8 liegen. Der Prozessor bricht nämlich bei Erkennen einer beliebigen anderen Taste sofort die Abfrage ab und ermittelt deren Ordnungszahl. Ob — neben Shift oder Control — eine weitere Taste in derselben Reihe betätigt wurde, ergibt der Vergleich mit $EF beziehungsweise $7F.

Die Flags Shift (oder Shift Lock) und Control spielen bei der Zuordnung der vorläufigen Tastennummer zu dem gewünschten ASCII-Code die entscheidende Rolle. Sie bestimmen, aus welcher der drei Tabellen (für die drei Belegungsebenen) der Tastencode entnommen wird. Jeder Taste kann ein beliebiger ASCII-Wert zugeteilt werden (und zwar für jede der drei Ebenen getrennt), indem man diesen ganz einfach an den Tabellenplatz setzt, der der Tastennummer entspricht. Bitte beachten Sie, daß die Tastennummer sich, wie beschrieben, aus den Matrix-Koordinaten ergibt. Sie hat nichts mit den im Bestückungsplan aufgeführten Nummern zu tun. Auf den Abdruck passender Tastencode-Tabellen möchten wir aus Platzgründen verzichten, zumal die Cherry-Tastatur eine beliebige Anordnung der Funktions- und Steuertasten durch Umstecken der Tastenkappen erlaubt.

Tabelle 3 zeigt die Initialisierungssequenz des Terminal-Programms, in welcher der beim Reset eingenommene Betriebszustand festgelegt wird. Änderungswünsche bezüglich Baudrate, Übertragungsformat und so weiter lassen sich leicht durch Einsetzen anderer Initialisierungs-Konstanten verwirklichen. Alle dazu erforderlichen Daten waren in den beiden Teilen der Projekt-Beschreibung in Tabellenform abgedruckt.

```
991B: A2 FF     RESET   LDX #$FF
991D: 9A                TXS             Stackpointer setzen
991E: 78                CLI             Kein Interrupt
991F: D8                CLD             Binär rechnen
9920: A9 00             LDA #0
9922: 85 14             STA MCR         Prozessor Mode Control
9924: A9 45             LDA #$45        Receiver, Edge Detect
9926: 85 12             STA IER         Interrupt Enable Register
9928: A9 C4             LDA #$C4        XMTR, RCVR Enable, Format
992A: 85 15             STA SCCR        Serial Communications
992C: A9 70             LDA #$70
992E: 85 57             STA OUTIND      Puffer-Aus-Zeiger
9930: A9 6F             LDA #$6F
9932: 85 56             STA ININD       Puffer-Ein-Zeiger
9934: A9 10             LDA #$10        Alpha Lock
9936: 85 40             STA KMODE       Keyboard Control
9938: A9 0A             LDA #$0A        Off Line, 9600 Baud
993A: 85 41             STA IFMODE      Interface Control
993C: A9 00             LDA #0          Auto Line Feed aus
993E: 85 43             STA BMODE
9940: A9 FE             LDA #%11110111  Drucker-Strobe aus
9942: 85 02             STA PC
```

Tabelle 3. Terminal-Initialisierung

Wer auf die Begrüßungsmeldung 'c't-Terminal...' verzichten möchte, sollte an den Speicherplätzen $994F bis 9953 (einschließlich) NOPs ($EA) einsetzen.

Verschiedene Anwendungs-Programme setzen ein Bildformat von 24 x 80 voraus. Dies stellt man durch folgende Änderungen ein: Adresse $9164 = $18 (Anzahl der dargestellten Zeilen), Adresse $9178 = $80 (L-Byte von Anzahl der dargestellten Zeichen).

Georg Werner

SuperTape für 6502-Computer

Vorgestellt am Beispiel von VC-20 und C64

Welcher VC-20-Besitzer hat sich nicht schon einmal gewünscht, daß sein Computer auch C64-Kassetten lesen könnte. Aber die unterschiedliche Aufzeichnungsgeschwindigkeit der beiden Rechner machte einen Programmtausch bisher unmöglich. Dieses Problem, das Commodore nicht lösen konnte (oder wollte), erledigt SuperTape. Darüber hinaus wird eine Voraussetzung zum Austausch von Daten mit Computern anderer Hersteller geschaffen, soweit diese SuperTape 'kennen'. Mit einer Übertragungsgeschwindigkeit von wahlweise 3600 oder 7200 Baud ist SuperTape außerdem bis zu sechsmal schneller als das Original-Verfahren.

Der Kassettenrecorder wird von den Commodore-Computern auf zwei verschiedene Arten benutzt. Zum einen mit den Befehlen SAVE und LOAD/VERIFY für Programme, zum anderen mit den Befehlen OPEN, PRINT# und INPUT# für Dateien. Bei der letzteren Aufzeichnungsart wird das File in Blöcke von 192 Byte zerhackt und auf das Band geschrieben. Die Anwendung von SuperTape ist hier kaum sinnvoll, weil der Zeitvorteil nicht mehr ins Gewicht fällt. Außerdem ist es nahezu unmöglich, die nur etwa zwei Zentimeter langen Blöcke auf dem Band wiederzufinden.

Für denjenigen, der mit Dateien arbeiten will, bietet sich ein anderes Verfahren an: Datenblöcke werden mittels LOAD in einen 4...8 KByte großen Puffer gebracht und können dort mit 'random access' bearbeitet werden. Anschließend wird der Block mittels SAVE zurückgeschrieben. Mit einem geeigneten 'Tape Operating System' und einem fernsteuerbaren Spulentonbandgerät lassen sich damit Dateien von bis zu 100 MByte verarbeiten (freilich wesentlich langsamer als beispielsweise mit Floppys als Massenspeicher).

Kleiner Eingriff

Doch nun zur Praxis: Zum Einbau von SuperTape in den VC-20 und in den C64 ist eine winzige Hardware-Ergänzung nötig. Der 'Read'-Eingang des Kassettenports ist nämlich mit dem FLAG-Eingang des Interface-Bausteins CIA 6526 verbunden, weil Commodore eine Interrupt-Steuerung für das Lesen verwendet. SuperTape benötigt aber eine Möglichkeit zur direkten Abfrage des Eingangspegels, denn das Retten der Register beim Interrupt würde zu lange dauern. Aus diesem Grund muß 'Read' beim VC-20 mit JOYØ und beim C64 mit PA2 verbunden werden. Wer diesen Eingang sonst anderweitig verwendet, sollte in die Verbindungsleitung einen Schalter einbauen (siehe Bild 1).

Um einen Bit-Strom mit 7200 Baud zu übertragen, muß die Programm-Laufzeit speziell für den C64 sorgfältig optimiert werden. Was der VC-20 mit seiner Taktfrequenz von 1,108 MHz noch 'spielend' schafft, wird für den 11 Prozent langsameren C64 schon zum Problem. Bei 7200 Baud darf das Senden eines Bits nur 139 µs dauern. In dieser Zeitspanne müssen für ein 0-Bit zwei Flankenwechsel erzeugt werden. Es stehen im schlechtesten Fall nur 50 Taktzyklen zur Verfügung. Besonders kritisch ist das 'Timing' zwischen zwei Bytes — hier reicht die Zeit nicht aus, um den Zeiger für die Übertragung zu erhöhen und das Ende der Sendung zu erkennen.

Verteilte Zeit

Es bleibt nichts anderes übrig, als die notwendigen Berechnungen zu verteilen. Das Programm sieht dann so aus:

Als erstes wird ein neues Byte aus dem Speicher geholt. Dann werden zunächst vier Bit übertragen. Jetzt prüft der Prozessor, ob das Ende des Datensatzes schon erreicht ist. Erst dann sendet er die restlichen vier Bits. Dieses Verfahren, eine Art 'Pipelining', ermöglicht es, die Übertragungsgeschwindigkeit einzuhalten. Ein Nachteil besteht darin, daß das Programm an Übersichtlichkeit verliert und weniger gut zu verstehen ist. Aus diesem Grund sollen die einzelnen Routinen hier etwas ausführlicher beschrieben werden.

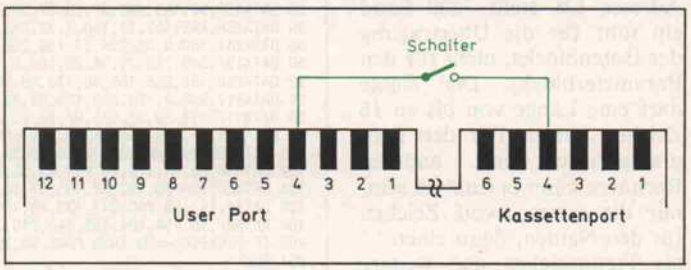


Bild 1. SuperTape-'Hardware'

Das Hauptprogramm SUPERSAVE übernimmt neben der Steuerung des Ablaufs noch das Übertragen des Programmnamens und der Parameter in den Kassettenpuffer.

Die Routine Write Block führt die gesamte Übertragung eines Datenblocks aus. Mit ihrer Hilfe werden sowohl der Parameter- als auch der Programmblock auf das Band geschrieben. Dazu ruft Write Block das Unterprogramm Write Byte auf, das den eigentlichen Sendevorgang ausführt. Write Block wird seinerseits vom Hauptprogramm aufgerufen, wobei das Programm zwischen zwei Aufrufen einige Zeit wartet, damit beim späteren Laden ausreichend Zeit bleibt, den Programmnamen zu analysieren.

Write Byte generiert die Pegeländerungen am Kassettenausgang, die zum Schreiben auf das Band erforderlich sind. Dabei stützt es sich auf die komfortablen 16-Bit-Timer der Commodore-Rechner.

Das Programmsegment Port Initial schaltet den Port für SAVE als Ausgang und startet den Kassettenmotor. Da der Videocontroller des C64 leider den gesamten Zeitablauf stört, wird der Bildschirm bei diesem Rechner in der Portroutine ausgeschaltet. Sämtliche Interrupts müssen gleichfalls abgeschaltet werden, denn SuperTape verträgt keine Unterbrechungen des Ablaufs. Das hat zur Folge, daß die Tastatur nicht mehr abgefragt wird. Das Programm läßt sich also mit der Stoptaste nicht mehr unterbrechen. Hier schafft die Routine NMI Abhilfe: Sie bewirkt, daß durch Betätigen der Restore-Taste ein BASIC-Warmstart eingeleitet wird.

LOAD und VERIFY gleichen sich im gesamten Programmablauf — mit Ausnahme der Tatsache, daß im einen Fall statt einer Schreiboperation ein Vergleich mit den im Speicher stehenden Daten ausgeführt wird. Es war deshalb sinnvoll, das Programm SUPERLOAD so anzulegen, daß wahlweise (in Abhängigkeit von einer Flag) beide Operationen ausgeführt werden können.

Ein Hauptprogramm übernimmt wieder die Steuerung und das Überprüfen des Filenamens. Die Daten werden in der Routine Readblock gelesen. Tabelle 1 veranschaulicht die Pipeline-Struktur dieses Programmteils.

Lesen (liest ein Byte vom Band)
2 Bit lesen
if LOAD gelesenen Wert im Speicher ablegen
2 Bit lesen
Vergleich mit Speicherinhalt bei Ungleichheit LOAD ERROR
2 Bit lesen
Incrementiere Zeiger
2 Bit lesen
if ENDE then END else GOTO LESEN

Tabelle 1. Struktur der Routine Readbyte.

READ macht von dem Unterprogramm BIT Gebrauch, das dazu dient, jeweils ein Bit vom Band zu lesen. Dabei wird, wenn es sich um ein 1-Bit handelt, gleichzeitig der Prüfsummen-Zähler inkrementiert. Das gelesene Bit wird jeweils von links in die Speicherzeile $57 'rotiert', die als Lesepuffer dient. Nach achtmaligem Aufruf von Bit enthält der Lesepuffer ein vollständiges Byte.

Die Synchronisation auf den 'Byte-Takt' der vom Band gelesenen Daten erfolgt so: Der Computer liest jeweils ein Bit und prüft dann, ob sich im Lesepuffer anschließend das 'SYN'-Zeichen $16 befindet. Wenn das der Fall ist, werden die weiteren Daten Byte-weise gelesen (Label RHA). Bis zur Abschlußmarke ($2A bzw. $C5) müssen jetzt (mindestens elf) weitere SYN-Zeichen folgen, sonst wird der Synchronisationsversuch wiederholt.

Eingebunden

Mit den Programmen SUPERLOAD und SUPERSAVE lassen sich Programme auf Kassette retten und von dieser lesen. Damit sie für den Benutzer bequem verfügbar sind, müssen die Programme noch in das Betriebssystem eingebunden werden. Glücklicherweise ist das bei den Commodore-Rechnern leicht zu machen, weil alle internen Funktionen über Vektoren aufgerufen werden. SuperTape kann durch Veränderung dieser Zeiger sehr einfach aufgerufen werden, wobei das Betriebssystem schon die Parameter wie Filenamen und Sekundäradresse bereitstellt.

Vom BASIC her kann SuperTape daher mit der Befehlsfolge

LOAD 'name', 9, SEC

aufgerufen werden. Eine Null als Sekundäradresse (SEC) bedeutet dabei, daß das Programm an den Anfang des BASIC-Speichers geladen wird. Mit einer Angabe ungleich Null wird das Programm an die Stelle geladen, die im Parameter-Block angegeben ist. Damit ist ein relokatives Laden von Maschinenprogrammen leicht möglich.

Einzelne Zeichen des Namens können bei LOAD, wie für das SuperTape-Verfahren gefordert, mit '?' maskiert werden. Auf die Option 'Maskierung des Restnamens' mit '*' wurde verzichtet. Statt dessen kann ein File mit dem Namen 'ABCDEFGHIJK' mit 'LOAD 'ABC'' oder auch 'LOAD 'A'' geladen werden. Auch das Lesen ohne Angabe eines Namens ist möglich (genau, wie vom Commodore-Betriebssystem her gewohnt).

Auf Band gespeichert wird ein Programm mit dem Kommando

SAVE 'name', 9, SEC.

Die Sekundäradresse 0 legt dabei die Baudrate 3600 fest, die Adresse 128 stellt 7200 Baud ein (nur für die Übertragung des Datenblocks, nicht für den Parameterblock). Der Name darf eine Länge von bis zu 16 Zeichen haben. Für den Programmtausch mit anderen Rechnern kann es sinnvoll sein, nur die ersten zwölf Zeichen für den Namen, dann einen '.' als Trennzeichen und weitere drei Buchstaben zur Typkennzeichnung zu benutzen. Dem VC-20 und dem C64 ist das egal, da sie den Typ wie den Namen behandeln.

Zum Schluß bleibt noch die Frage zu klären, in welchen Speicherbereich SuperTape geladen wird. Damit das Programm auf allen VC-20-Rechnern mit beliebiger Speicherausstattung läuft, wird es auf diesem im Bereich 1900...1D00 abgelegt. (Der Bereich von 1D00 bis zum Bildspeicher wird von dem in c't 4/84 beschriebenen Centronics-Schnittstellenprogramm beansprucht.) Besitzer von Speichererweiterungen können das Programm entweder ändern oder den Speicher mit

POKE 44,32:
POKE 32*256,0:NEW

für diese Anwendung freihalten. Beim C64 liegt das Programm ab der Adresse C000.

Da jede Betätigung der Restore-Taste alle Vektoren des Betriebssystems wieder auf den Anfangswert zurücksetzt, muß SuperTape anschließend wieder in das Betriebssystem eingebunden werden. Dies erfolgt beim VC-20 mit

```
POKE 816, 181:
POKE 817,26:
POKE 818,0: POKE 819,25
```

und beim C64 mit

```
POKE 816,179:
POKE 817,193:
POKE 818,0:
POKE 819,192.
```

Wird dies vergessen, meldet sich der Rechner mit einem 'device not present error'.

SuperTape läuft zwar auf dem VC-20 in der Grundversion. Da aber nur noch 2,3 KByte RAM frei sind, paßt der BASIC-Lader nicht in den Speicher. Es muß deshalb zum Laden ein anderes Verfahren verwendet werden.

Auf den Abdruck eines BASIC-Laders für den C64 verzichten wir hier aus Platzgründen. Falls kein Assembler zur Verfügung steht, müssen Sie das Programm wohl oder übel 'von Hand' assemblieren. Bitte beachten Sie, daß für den C64 einige Programmteile ausgetauscht werden müssen.

```
10 REM**************************************
11 REM       SUPERTAPE   VC-20
12 REM**************************************
13 POKE816,181:POKE817,26      :REM LOAD VEKTOR
14 POKE818,0  :POKE819,25      :REM SAVE VEKTOR
20 FORI= 6400 TO 7217 :READDC:POKEI,DC:NEXT
22 DATA165,186,201,9,240,3,76,133,246,32,170,26,160,0,196,183,240,8,177
24 DATA187,153,60,3,200,208,244,169,0,153,60,3,200,192,16,144,246,165
26 DATA185,141,76,3,165,43,141,77,3,165,44,141,78,3,56,165,45,229,43,141
28 DATA79,3,165,46,229,44,141,80,3,169,0,141,81,3,141,82,3,141,83,3,141
30 DATA84,3,32,71,26,169,34,133,87,169,135,133,88,169,1,133,90,32,183
32 DATA248,32,40,247,32,86,26,169,42,32,160,25,162,255,160,255,136,208
34 DATA253,202,208,248,165,185,16,12,169,136,133,87,169,59,133,88,169
36 DATA0,133,90,165,43,133,91,165,44,133,92,165,45,133,93,165,46,133,94
38 DATA169,197,32,160,25,24,76,126,26,72,169,64,133,95,169,22,32,237,25
40 DATA32,237,25,198,95,208,244,104,32,237,25,32,237,25,160,0,132,96,132
42 DATA97,177,91,32,237,25,230,91,208,2,230,92,32,237,25,165,91,197,93
44 DATA208,236,165,92,197,94,208,230,165,97,72,165,96,32,237,25,32,237
46 DATA25,104,32,237,25,32,237,25,76,237,25,133,98,160,4,70,98,176,3,166
48 DATA88,44,166,87,173,29,145,41,32,240,249,142,24,145,176,3,169,0,44
50 DATA165,90,141,25,145,173,32,145,73,255,141,32,145,176,29,173,29,145
52 DATA41,32,240,249,142,24,145,169,0,141,25,145,173,32,145,73,255,141
54 DATA32,145,136,208,191,165,98,96,165,96,105,0,133,96,165,97,105,0,133
56 DATA97,136,208,173,165,98,96,169,60,133,91,169,3,133,92,133,94,169
58 DATA85,133,93,96,120,169,0,141,27,145,169,255,141,34,145,169,0,141
60 DATA19,145,169,1,141,24,145,141,25,145,169,253,141,28,145,162,255,160
62 DATA255,136,208,253,202,208,248,96,162,255,142,28,145,162,173,142,24
64 DATA3,162,254,142,25,3,166,91,164,92,88,96,169,255,141,28,145,169,173
66 DATA141,24,3,169,254,141,25,3,88,24,32,249,253,108,2,192,169,147,141
68 DATA24,3,169,26,141,25,3,96,133,147,169,0,133,144,165,186,201,9,240
70 DATA3,76,79,245,32,148,248,32,71,246,32,86,26,165,147,133,254,169,0
72 DATA133,147,32,170,26,32,71,26,169,216,133,90,169,42,32,78,27,160,99
74 DATA32,230,241,160,0,185,60,3,32,210,255,200,192,16,208,245,160,0,196
76 DATA183,240,17,177,187,217,60,3,208,3,200,208,242,201,63,208,203,200
78 DATA208,235,165,254,133,147,32,106,246,165,43,133,91,165,44,133,92
80 DATA165,185,240,10,173,77,3,133,91,173,78,3,133,92,24,173,79,3,101
82 DATA91,133,93,173,80,3,101,92,133,94,173,76,3,48,3,169,216,44,169,79
84 DATA133,90,169,197,32,78,27,24,76,126,26,120,133,89,32,235,27,201,22
86 DATA208,249,162,11,160,8,32,235,27,136,208,250,201,22,208,235,202,208
88 DATA241,160,8,32,235,27,136,208,250,201,22,240,244,197,89,240,7,201
90 DATA197,240,212,76,34,28,169,0,133,96,133,97,160,8,32,235,27,136,208
92 DATA250,133,255,165,96,133,98,165,97,133,99,32,235,27,32,235,27,165
94 DATA147,208,4,165,255,145,91,32,235,27,32,235,27,165,255,209,91,240
96 DATA3,76,34,28,32,235,27,32,235,27,230,91,208,2,230,92,32,235,27,32
98 DATA235,27,133,255,165,92,197,94,208,193,165,91,197,93,208,187,165
100 DATA255,197,98,208,12,160,8,32,235,27,136,208,250,197,99,240,3,76
102 DATA34,28,96,173,29,145,41,32,240,249,173,17,145,41,4,197,88,240,1
104 DATA24,133,88,102,87,165,87,16,6,230,96,208,2,230,97,173,17,145,41
106 DATA4,197,88,240,247,133,88,165,90,141,24,145,169,0,141,25,145,165
108 DATA87,96,104,104,165,147,240,3,169,28,44,169,29,56,76,126,26,0
200 IF PEEK(56)<=30 THEN POKE 56,25
201 NEW
```

```
*       ***************
*       SUPER SAVE FUER VC-20
*       ***************
        ORG $1900
LSL1    EQU 34
LSL0    EQU 135
HSL1    EQU 136
HSL0    EQU 59
IRQ     EQU $911D
TIMER   EQU $9118
STW     EQU $00
TSTART  EQU $9119
NMIV    EQU $FEAD
PRESSR  EQU $F8B7
SAVING  EQU $F728
PORTAT  EQU $9120
BETRSA  EQU $F685
        LDA $BA;GERAET
        CMP #$09
        BEQ *+3
        JMP BETRSA;KEIN SUPERTAPE
        JSR NMI;STOP HERSTELLEN
        LDY #$00
NAME    CPY $B7    ;UEBERTRAEGT NAME IN PUFFER
        BEQ FUELL
        LDA ($BB),Y
        STA $033C,Y
        INY
        BNE NAME
FUELL   LDA #$00   ;FUELLT NAME AUF 16 ZEICHEN
        STA $033C,Y
        INY
        CPY #$10
        BCC FUELL
        LDA $B9    ;SECADR. IN PUFFER
        STA $034C
        LDA $2B    ;START IN PUFFER
        STA $034D
        LDA $2C
        STA $034E
        SEC        ;LAENGE IN PUFFER
        LDA $2D
        SBC $2B
        STA $034F
        LDA $2E
        SBC $2C
        STA $0350
        LDA #$00
        STA $0351  ;4 LEERSTELLEN
        STA $0352
        STA $0353
        STA $0354
        JSR PUFFER
        LDA #LSL1  ;3600 BAUD
        STA $57
        LDA #LSL0
        STA $58
        LDA #$01
        STA $5A
        JSR PRESSR
        JSR SAVING
        JSR PORT
        LDA #$2A
        JSR WRITE
        LDX #$FF
A       LDY #$FF
B       DEY
        BNE B
        DEX
        BNE A
        LDA $B9    ;BAUDRATE FUER DATEN
        BPL POINT
        LDA #HSL1
        STA $57
        LDA #HSL0
        STA $58
        LDA #$00
        STA $5A
POINT   LDA $2B    ;SETZT ZEIGER
        STA $5B
        LDA $2C
        STA $5C
        LDA $2D
        STA $5D
        LDA $2E
        STA $5E
        LDA #$C5
        JSR WRITE
        CLC
        JMP ENDE
*       ***************
*       WRITE BLOCK
*       ***************
WRITE   PHA
        LDA #$40
        STA $5F
SYNC    LDA #$16   ;UEBERTRAEGT SYN
        JSR WBYTE
        JSR WBYTE
        DEC $5F
        BNE SYNC
        PLA
        JSR WBYTE
        JSR WBYTE
        LDY #$00   ;PRUEFSUMME = 0
        STY $60
        STY $61
WDAT    LDA ($5B),Y
        JSR WBYTE
        INC $5B
        BNE *+2
        INC $5C
        JSR WBYTE
        LDA $5B
        CMP $5D
        BNE WDAT
        LDA $5C
        CMP $5E
        BNE WDAT
        LDA $61    ;PRUEFSUMME AUF BAND
        PHA
        LDA $60
        JSR WBYTE
        JSR WBYTE
        PLA
        JSR WBYTE
        JSR WBYTE
        JMP WBYTE
*       ***************
*       WRITE BYTE
*       ***************
WBYTE   STA $62
        LDY #$04
WLOOP   LSR $62
        BCS *+3
        LDX $58
        DFB $2C
        LDX $57
        LDA IRQ    ;WARTET AUF TIMEOUT
        AND #$20
        BEQ *-7
        STX TIMER
        BCS *+3
        LDA #$00
        DFB $2C
        LDA $5A
        STA TSTART
        LDA PORTAT
        EOR #$ FF
        STA PORTAT
        BCS WPRUEF
        LDA IRQ
        AND #$20
        BEQ *-7
        STX TIMER
        LDA #STW
        STA TSTART
        LDA PORTAT
        EOR #$FF
        STA PORTAT
        DEY
        BNE WLOOP
        LDA $62
        RTS
WPRUEF  LDA $60
        ADC #$00
        STA $60
        LDA $61
        ADC #$00
        STA $61
        DEY
        BNE WLOOP
        LDA $62
        RTS
*       ***************
*       PUFFERZEIGER
*       ***************
PUFFER  LDA #$3C
        STA $5B
        LDA #$03
        STA $5C
        STA $5E
        LDA #$55
        STA $5D
        RTS
*       ***************
*       PORT INITIAL
*       ***************
PORT    SEI
        LDA #$00
        STA $9118
        LDA #$FF
        STA $9122
        LDA #$00
        STA $9113
        LDA #$01
        STA $9118
        STA $9119
        LDA #$FD
        STA $911C
        LDX #$FF
C       LDY #$FF
D       DEY
        BNE D
        DEX
        BNE C
        RTS
*       ***************
*       ENDE
*       ***************
ENDE    LDX #$FF
        STX $911C
        LDX #>NMIV
        STX $0318
        LDX #<NMIV
        STX $0319
        LDX $5B
        LDY $5C
        CLI
        RTS
*       ***************
*       BRK
*       ***************
BRK     LDA #$FF
        STA $911C
        LDA #>NMIV
        STA $0318
        LDA #<NMIV
        STA $0319
        CLI
        CLC
        JSR $FDF9
        JMP ($C002)
*       ***************
*       NMI
*       ***************
NMI     LDA #>BRK
        STA $0318
        LDA #<BRK
        STA $0319
        RTS
*       ***************
*       SUPER LOAD VC-20
*       ***************
        STA $93
        LDA #$00
        STA $90    ;STATUS LOESCHEN
        LDA $BA    ;GERAET
        CMP #$09
        BEQ *+3
        JMP $F54F  ;KEIN SUPERTAPE
        JSR PRESSP
        JSR SEARCH
        JSR PORT
        LDA $93
        STA $FE
        LDA #$00
        STA $93
HEAD    JSR NMI
        JSR PUFFER
        LDA #LOSPED
        STA SPEED
        LDA #$2A   ;HEADER
        JSR READ
        LDY #$63
        JSR FOUND
        LDY #$00
PRINT   LDA $033C,Y;DRUCKT NAMEN
        JSR $FFD2
        INY
        CPY #$10
        BNE PRINT
        LDY #$00
VERGL   CPY $B7    ;VERGLEICH DER NAMEN
        BEQ OK
        LDA ($BB),Y
        CMP $033C,Y
        BNE ???
        INY
        BNE VERGL
???     CMP #$3F
        BNE HEAD
        INY
        BNE VERGL
OK      LDA $FE
        STA $93
        JSR TEXT;LOADING/VERIFY
        LDA $2B    ;SETZT POITER
        STA $5B
        LDA $2C
        STA $5C
        LDA $B9
        BEQ ORGIN;LADEN BASIC/MASCHPROG
        LDA $034D
        STA $5B
        LDA $034E
        STA $5C
ORGIN   CLC
        LDA $034F  ;BERECHNUNG FILEENDE
        ADC $5B
```

```
        STA $5D
        LDA $0350
        ADC $5C
        STA $5E
        LDA $034C
        BMI *+3     ;BAUDRATE
        LDA #LOSPED
        DFB $2C
        LDA #HISPED
        STA SPEED
        LDA #$C5    ;DATENBLOCK
        JSR READ
NC      CLC
        JMP ENDE
*       ****************
*       READ BLOCK
*       ****************
READ    SEI
        STA $59
READL   JSR BIT
        CMP #$16
        BNE READL
        LDX #$0B
RHA     LDY #$08
        JSR BIT
        DEY
        BNE *-6
        CMP #$16
        BNE READL
        DEX
        BNE RHA
RHB     LDY #$08
        JSR BIT
        DEY
        BNE *-6
        CMP #$16
        BEQ RHB
        CMP $59
        BEQ LLL
        CMP #$C5
L       BEQ READL
        JMP ERROR
LLL     LDA #$00    ;PRUEFSUMME=0
        STA $60
        STA $61
        LDY #$08
        JSR BIT
        DEY
        BNE *-6
        STA $FF
RBYTE   LDA $60
        STA $62
        LDA $61
        STA $63
        JSR BIT
        JSR BIT
        LDA $93
        BNE VERI
        LDA $FF
        STA ($5B),Y
VERI    JSR BIT
        JSR BIT
        LDA $FF
        CMP ($5B),Y
        BEQ *+3
        JMP ERROR
        JSR BIT
        JSR BIT
        INC $5B
        BNE *+2
        INC $5C
        JSR BIT
        JSR BIT
        STA $FF
        LDA $5C
        CMP $5E
        BNE RBYTE
        LDA $5B
        CMP $5D
        BNE RBYTE
        LDA $FF
        CMP $62
        BNE LOER
        LDY #$08
        JSR BIT
        DEY
        BNE *-6
        CMP $63
        BEQ *+3
LOER    JMP ERROR
        RTS
*       ****************
*       READ BIT
*       ****************
BIT     LDA $911D
        AND #$20
        BEQ BIT
        LDA $9111
        AND #$04
        CMP $58
        BEQ REINS
        CLC
REINS   STA $58
        ROR $57
        LDA $57
        BPL FLAN
        INC $60
        BNE FLAN
        INC $61
FLAN    LDA $9111
        AND #04
        CMP $58
        BEQ FLAN
        STA $58
        LDA $5A
        STA $9118
        LDA #$00
        STA $9119
        LDA $57
        RTS
*       ****************
*       ERROR
*       ****************
ERROR   PLA
        PLA
        LDA $93
        BEQ *+3
        LDA #28
        DFB $2C
        LDA #29
        SEC
        JMP ENDE
WER     BRK
```

FÜR DEN C 64 MÜSSEN FOLGENDE MODULE AUSGETAUSCHT WERDEN

```
*       ****************
*       SUPER SAVE FUER C 64
*       ****************
        ORG $C000
LSL1    EQU 255
LSL0    EQU 118
HSL1    EQU 120
HSL0    EQU 52
IRQ     EQU $DD0D
TIMER   EQU $DD04
STW     EQU $D9
TSTART  EQU $DD0E
NMIV    EQU $FE47
PRESSR  EQU $F838
SAVING  EQU $F68F
PORTAT  EQU $01
BETRSA  EQU $F5ED
*       ****************
*       WRITE BYTE
*       ****************
WBYTE   STA $62
        LDY #$04
WLOOP   LSR $62
        BCS *+3
        LDX $58
        DFB $2C
        LDX $57
        LDA IRQ     ;WARTET AUF TIMEOUT
        AND #$01
        BEQ *-7
        STX TIMER
        LDA #STW
        STA TSTART
        LDA PORTAT
        EOR #$08
        STA PORTAT
        BCS WPRUEF
        LDA IRQ
        AND #$01
        BEQ *-7
        STX TIMER
        LDA #STW
        STA TSTART
        LDA PORTAT
        EOR #$08
        STA PORTAT
        DEY
        BNE WLOOP
        LDA $62
        RTS
WPRUEF  LDA $60
        ADC #$00
        STA $60
        LDA $61
        ADC #$00
        STA $61
        DEY
        BNE WLOOP
        LDA $62
        RTS
*       ****************
*       PORT INITIAL
*       ****************
PORT    SEI
        LDA $01
        AND #$1F
        STA $01
        LDA #$00
        STA $DD03
        STA $DD05
        LDA #$FF
        STA $DD04
        LDA #STW
        STA $DD0E
        LDX #$FF
C       LDY #$FF
D       DEY
        BNE D
        DEX
        BNE C
        LDA $D011
        AND #$EF
        STA $D011
        RTS
*       ****************
*       ENDE
*       ****************
ENDE    LDX $01
        ORA #$20
        STX $01
        LDX #>NMIV
        STX $0318
        LDX #<NMIV
        STX $0319
        LDX $5B
        LDY $5C
        CLI
        LDA $D011
        ORA #$10
        STA $D011
        RTS
*       ****************
*       BRK
*       ****************
BRK     LDA $01
        ORA #$20
        STA $01
        LDA #>NMIV
        STA $0318
        LDA #<NMIV
        STA $0319
        CLI
        LDA $D011
        ORA #$10
        STA $D011
        CLC
        JSR $FDF9
        JMP ($A002)
*       ****************
*       SUPER LOAD C 64
*       ****************
PRESSP  EQU $F817
SEARCH  EQU $F5AF
FOUND   EQU $F12F
TEXT    EQU $F5D2
LOSPED  EQU 181
HISPED  EQU 79
SPEED   EQU $DD04
BETRLO  EQU $F4AB
*       ****************
*       READ BIT
*       ****************
BIT     LDA $DD0D
        AND #$01
        BEQ BIT
        LDA $DD00
        AND #$04
        CMP $58
        BEQ REINS
        CLC
REINS   STA $58
        ROR $57
        LDA $57
        BPL FLAN
        INC $60
        BNE FLAN
        INC $61
FLAN    LDA $DD00
        AND #04
        CMP $58
        BEQ FLAN
        STA $58
        LDA #STW
        STA TSTART
        LDA $57
        RTS
*******************************
```

bst

Produkte für den

Commodore CBM-64

MODULE		SPIELE	
Forth	198,–	Golf 64 (neu!)	49,–
Super MON	159,–	Super Catha Trooper (neu!)	49,–
Super AID	149,–	**HARDWARE**	
Grafik Modul	179,–	Light Pen	85,–
SOFTWARE		Sprite Pen	159,–
A-Calc (Disk/Cas.)	136,–	IEEE 488	228,–
Wordpro 3 + / 64	285,–	5fach Modul Steckplatz	198,–
Home Manager (Disk/Cas.)	126,–	ROM Board	98,–
Super Base 64	420,–	Joy Stick "Radiant"	90,–
Sprite Editior 64 +	49,–	**BÜCHER**	
Grafik a.d. 64er (Disk/Cas.)	54,–	Programmers Ref. Guide	75,–
Compactor	59,–	Systemhandbuch	74,–
Music Pen	49,–	Assemblerkurs f. Anf.	38,–
		Getting the most out of the 64 (Jim Butterfield u.a.)	54,–
		64 Exposed	45,–

bst computronic
Burgstr. 126a
6000 Frankfurt 60

Tel.: 0611/453857 ** neues Info gegen 2,– DM Briefmarken!

Für den Computer zu Hause ein Möbelstück, das in jede Wohnung paßt.

Nur unser Computertisch bietet diese Vorzüge und Möglichkeiten:
1. Geräte verschließbar
2. Geräte auf stabilen Auszügen herausziehbar
3. kein lästiges Auf- und Abbauen, sofortige Einsatzmöglichkeit des Computers
4. Auszugboden für Drucker mit Problemlösung für Papiereinzug und Papieraustritt
5. fahrbar und mit Steckdosenleiste

Farben: Weiß und Braun

Ausführungen:
1. wie Abbildung
2. oder mit 2 Glastüren

Preis: Ausf. 1 ..... **448,— DM** Ausf. 2 ..... **470,— DM**

Lieferung frei Haus per Nachnahme. Maße: 90/80/50 cm (H/B/T)

Bezugsquelle:
ESH-Handelsagentur, Am Flutgraben 2, 5239 Hirtscheid

Floppy Disk Laufwerke

Firma / Modell	TEAC halbe Bauhöhe 5¼" 55A	55B	55E	55F	55G	TEAC 30A	BASF 6128	6138	6105
5=5" 8=8" 3=3"	5	5	5	5	5	3	5	5	8
Kapazität unformatiert KB	250	500	500	1000	1604	250	500	1000	1600
Anzahl der Köpfe	1	2	1	2	2	1	2	2	2
Anzahl Spuren pro Seite	40		80			40	80		77
Preise inclusive MwSt.	640	760	750	890	1040	490	665	795	1195

Drucker: **MANNESMANN/TALLY MT80** mit 7 Zeichensätzen, Linien- und Blockgrafik, direkte Nadelansteuerung, wegoptimierter bidirektionaler Druck und Selbsttest. Centronics-Schnittstelle (8 Bit parallel) **nur 980 DM incl. MwSt.**

Lasar Ilze Proficomputer mit 2 CPUs (Z80 und 6502) und 64 KB RAM auf dem Board. Tastatur mit 10er Block (Tasten sind mit 2 Funktionen belegt. Schaltnetzteil mit 7,5 A. Diesen PROFI:apple-kompatiblen und CP/M-fähigen Computer erhalten Sie für **1490 DM**

Ferner liefern wir: Winchester-Laufwerke (BASF)
Wechsel-Winchester (DRI)
Einplatinen-Computer (Doppeleuropakarte) sowie reichliches EDV-Zubehör

Alle Preise inclusive Mehrwertsteuer!

Gerhard Siemens Micro-Computer Service
Lenbachstr. 115, 7000 Stuttgart 1, Tel. (07 11) 85 90 88

MICROCOMPUTER

ATARI 600XL
ATARI 800XL

MÜNZENLOHER GMBH

Für dieses Computersystem gibt es bereits heute über 2000 Programme !

Wir sind autorisierter Händler für die gesamte ATARI-Computer-Palette. Guter Geräteservice vorhanden. Groß-Servicestelle von Atari in nächster Nähe.

Die neue SUPERPALETTE von ATARI

ATARI 600XL, 16K RAM, 24K ROM, (Inkl. ATARI-BASIC), Netzgerät, dt. Dokumentation **549,– DM**

ATARI 64 MODUL, Speichererweiterung für ATARI 600XL **349,– DM**

Technische Daten wie unten.

ATARI 800XL, 64K RAM, inkl. ATARI-Basic, Netzgerät, dt. Beschreibung **899,– DM**

ATARI 1050 Diskettenstation 999,– DM

Lernset 1010 (incl. Recorder, 2 Lernprogr., kompl. mit Anschluß (Netzteil, Kabel) **249,–**

ATARI 1027 Drucker 899,– DM
Schönschreibdrucker mit Interface u. Kabel.

ATARI 1025 Matrixdrucker 1.399,– DM
Incl. Kabel, Netzteil, Interface

ATARI 1020 Farbdrucker 799,– DM
Dieser Drucker ermöglicht Superfarbgraphik. Incl. Interface, Kabel, Netzt., Progr. Cassette.

Wir führen die gesamte EPSON-Drucker-Palette

Track-Ball f. schnelles präzises Spielen **179,–**

Monitore

BMC-BM-12 EN grün, 12", 20 MHz	398,–
BMC-BM-12 EY bernstein, w. oben	419,–
Farbmonitor Sanyo CD 3185 mit Grünschaltung	999,–
Farbmonitor Sharp DV 1400 6', 36 cm Bild, Fernbed., Video u. VHF-Eing.	1.049,–
Farbmonitor Sharp DV 1600 6', 42 cm Bild, Fernbed., Video u. VHF-Eing.	1.149,–

Staubschutzhauben

Formschön, Maßgeschneidert, Unzerreißbar

Atari 400	39,–	Atari 600XL	29,–
Atari 800	39,–	Atari 800XL	29,–
Atari 810 Disk	29,–	Atari 1050 Disk	29,–
Atari 410 Cass.	29,–	Atari 1010 Cass.	29,–
EPSON Drucker MX 80/82			39,–
EPSON Drucker RX 80/FX 80			39,–
EPSON Drucker MX 100/FX 100			49,–

Drucker Interface

Neues Interface von MPP 1150 mit eigenen Prozessor, Kabel und Stecker, läuft auf Visicalc, Text Wizard, Filemanager. Läuft ohne Software, auch Grafik möglich **395,–**

Neues billiges Interface für ATARI 248,–

Recorder Interface für ATARI 95,–

32K RAM Board ATARI 800 175,–

48K RAM Board ATARI 400 275,–

64K RAM Board ATARI 600 XL 298,–

Das neueste von der Firma OSS
Super Software in a SuperCartridge !

BASIC XL 395,– DM
Ein neues verbessertes BASIC, das viele Utilities beinhaltet.

ACTION 395,– DM
Die neue superschnelle Computersprache. Ähnlich wie FORTH.

MAC/65 395,– DM
Ein Supermacroassembler – Edit, Assemble, and Debug Quickly.

Weitere interessante Bücher für ATARI

ATARI BASIC (deutsch)
Das ideale Buch für den Computer-Neuling mit vielen Programmbeispielen **39,– DM**

Das große Spielebuch für ATARI 600/800XL
Programme, Tips u. Tricks. **29,80 DM**

Mein ATARI-Computer 59,– DM

First Book of ATARI	55,– DM
Second Book of ATARI	55,– DM
Third Book of ATARI	55,– DM
First Book of ATARI Graphics	55,– DM
First of ATARI Games	55,– DM
Mapping the ATARI	58,– DM
Inside ATARI DOS	85,– DM
The ATARI BASIC Sourcebook	55,– DM
Machine Language for Beginners	49,– DM
Home Energy Applications	58,– DM
First Book of Robots a. Computers	19,– DM

Tölzer Straße 5
D–8150 Holzkirchen / Obb.
Telefon: (0 80 24) 18 14
Gerätebau, Computersysteme – Software Lieferung per NN od. Vorkasse auf Postscheck-Kto. 2845 58–807 München od. Euroscheck.

NEU ★ NEU ★ NEU
für SHARP 700 Serie

Sonderangebot:
SHARP 711 mit Assembler oder Pascal u. weiteren Spielprogrammen **798,– DM**
SHARP 721 mit Recorder **998,– DM**
SHARP MZ-731 1.395,– DM
Printer-Interface MZ-7UPI 365,– DM
Floppy-Interface MZ7-FIB 595,– DM
Floppy-Basic-Interpreter MZ-7FBAS 145,–
Single-Floppy-Drive-MZ-7SFL 1.695,– DM
Doppel-Floppy-Drive MZ-7DFL 2.300,– DM
Formschöner **SHARP-Monitor MZ-1D05** RGB-Eingang m. brillanten Farbbild **1.448,–**
PCG-Graphiksystem (Modul) **360,– DM**

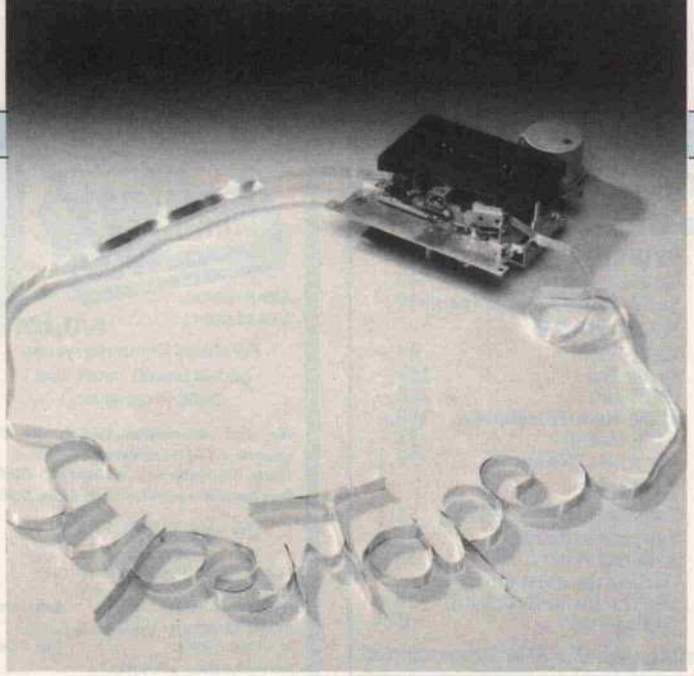
Software für SHARP MZ-700

Lagerprogramm-MZ-700 (C) 199,50 DM
Programm verwaltet 500 Artikel
Lager & Rechnung MZ-700 (C) 199,50 DM
Lager- u. Rechn.-Progr., div. Kundenadr.-Eing.
Scheckdruck MZ-700 (C) 199,50 DM
Div. Adress- u. Rechungseingabe, Aufschl.
Adressenverwaltung MZ-700 (C) 199,50 DM
Warenstatistik MZ-700 (C) 199,50 DM
Für 35 Warengruppen m. Kalkulationsprogr.
Terminverwaltung MZ-700 (C) 199,50 DM
universell einsetzbares Programm f. Verw.
Info-Datei MZ-700 (C) 199,50 DM
Informationen werden erfaßt, beim Suchlauf nach verschiedene Suchkriterien aufgelistet.
Fakturier-Automat MZ-700 (C) 199,50 DM
Programm ermöglicht Rechnungsschreiben m. div. Kunden und diversen Artikeln.
Mathematik MZ-700 (C) 199,50 DM
Statik (1) Durchlaufträger (C) 648,– DM
Statik (2) Rechteckplatte (C) 342,– DM
Statik (3) Stahlliste (C) 456,– DM
Statik (4) Pfahllasten (C) 912,– DM
Statik (5) Bemessung (C) 513,– DM
Werbung & Grafik (C) 199,50 DM
Bildschirmwerbung u. zus. Grafik-Druckprogr.
Convert MZ-80/A to MZ-700 (C) 199,50 DM
Maschinenprogramm convertiert Basic-Progr. von MZ-80/A auf den MZ-700

Weitere umfangreiche Diskettenversionen für den SHARP MZ-700, für 1 oder 2 Laufwerke:

Kunden/Adresssen-Programm	296,40 DM
Faktura-Programm	296,40 DM
Lager-Programm	296,40 DM
Mahnwesen	296,40 DM
Text-Verwaltung	296,40 DM
Statistik	296,40 DM

Über 100 Spielprogramme zwischen 20,– und 100,– DM stehen f. d. SHARP zur Verfügung.
Pascal 700 185,– DM
Real Interpreter. Befehlss. m. Bildsch. Editor.
Real-BASICCOMPILER 160,– DM
BASIC-Programme bis zu 50 mal schneller.
6502-Betriebssystem 100,– DM
Ma.-Bef. d. 6502 a. Ihren MZ-700 verarbeiten.
FORTRAN REAL COMPILER II 95,– DM
Sie erzeugen nach Compilerung ein reines Ma.-Progr., das bis zu 50 mal schneller ist.
SUPERDISASSEMBLER 75,– DM
eignet sich zum Ändern, Dokumentieren und Schreiben von komplexen Programmen.
Erweitertes BASIC 700 100,– DM
noch zusätzlich IF-THEN-ELSE, WHILE-WEND, REPEAT-UNTIL (nur 21K).
Deutsches S-Basic 40,– DM
Dt. Umlaute direkt von der Tastatur.
AS-700 Assembler / Editor 129,– DM
EM-700 Erweiterter Monitor 75,– DM
Daten-Bank DPB-701+ 185,– DM
Ersetzt Karteikasten, Adresskarteien, usw. (500 Adressen oder 1000 Artikel).
700-Kundenverwaltung 100,– DM
Verw. 400 Kunden m. Adr. u. Konten.
700-Minitext 100,– DM
Alle Schriftarten, Farben u. Grafikfunktionen in einem Textprogramm.
700-Tortengrafik 50,– DM
700-Kurvendiskussion 30,– DM
TR-700 Taschenrechner 50,– DM
Bedienbar wie ein wissenschaft. Taschenr.
Wecker-700 30,– DM
Weckt Sie pünktlich m. angenehmer Melodie.



SuperTape-Komfort für den ZX 81

Andreas Stiller

Die im vorigen Heft vorgestellten SuperTape-Basisroutinen benötigen einen 'gefüllten' Parameter-Puffer, der ihnen alle notwendigen Informationen zur Verfügung stellt. Dies kann natürlich ein BASIC-Programm gewährleisten, das die gewünschten Parameter in den Pufferbereich 'POKE'. Komfortabler ist aber das hier vorgestellte Maschinenprogramm, das überdies einen SuperTape-Kaltstart erlaubt.

Die Bedienungsbefehle beanspruchen nur eine BASIC-Zeile und können sowohl direkt als auch aus dem Programm heraus angesprochen werden. Sie entsprechen also in ihrer Handhabung den Standardbefehlen. Um allerdings der gut funktionierenden Sinclair-Syntaxkontrolle zu entkommen, sind sie in einen String gepackt, der mit dem Aufruf

```
PRINT STR$ USR ZZ =
'Befehlsstring'
```

abgearbeitet wird (ZZ soll die Aufrufadresse 31415 enthalten). Der String kann auch in einer Stringvariablen enthalten sein.

Die Befehle haben dabei folgende Struktur:

```
'MODE,NAME.TYP,
 START,ENDE'
```

Der Mode kennzeichnet die Aufgabe:

S für SAVE mit 3600 Bd
Q für QUICKSAVE mit 7200 Bd
L für LOAD
V für VERIFY (nur für Hexfiles)

Falls Sie wünschen, den nun folgenden Namen (bis zu zwölf Zeichen) und den Typ (bis zu drei Zeichen) in ASCII auszugeben, können Sie dies durch ein angehängtes U (also 'SU' beziehungsweise 'QU') anzeigen. Schließlich kann Mode auch durch ein 'N' ergänzt werden, was die Bildschirmausgabe der Parameter verhindert — zum Beispiel, um einen geretteten Bildschirm nicht zu überblenden.

Der Name sollte die Zeichen

```
. , / * ?
```

nicht beinhalten, da diese als Trenn- oder Jokersymbole interpretiert werden.

Der Typ kennzeichnet die Art des zu übertragenden Datenblocks (File). Die Typ-Bezeichnung kann man frei wählen, allerdings gibt es einige Standardtypen, die das Programm erkennt, und woraufhin es spezielle Routinen anspringt.

Der Typ BAS bedeutet ein komplettes BASIC-Programm, mit Systemvariablen, Programm, Bildschirm und Variablenfeld. Daher 'weiß' der Rechner, daß die Startadresse 4009h beträgt und und daß die Systemvariable E-Line auf das Ende zeigt.

Eine Besonderheit stellt der Typ KAL dar. Hierdurch wird der Rechner veranlaßt, ein Kaltladeprogramm auszugeben. Im 'kalten' Zustand, also nach dem Einschalten, kennt der Rechner ja noch keine Programme, außer seinen ROM-Routinen. Somit müssen die SuperTape-Laderoutinen auch erst einmal geladen werden. Das kann allerdings nur mit dem normalen Sinclair-Verfahren geschehen. Leider kennt der ZX 81 keine speziellen Ladebefehle für Maschinenprogramme. Diese werden üblicherweise in REM-Statements untergebracht, von wo aus sie dann ein Verschiebeprogramm in den gewünschten Adreßbereich bringen kann.

Das hier vorgestellte Kaltstartprogramm übernimmt diese Aufgabe, indem es

erst die SuperTape-Laderoutine lädt — ohne den Bildschirm unnötigerweise mitzuladen —,

RAMTOP und den Stackpointer neu setzt, um den darüberliegenden Adreßbereich zu schützen,

dann auf SuperTape umschaltet und die restlichen Super Tape-Routinen und eventuell eigene Maschinenprogramme in diesen Adreßbereich lädt,

und schließlich das BASIC-Programm — natürlich auch in SuperTape — vom Band liest.

Diese letzten beiden Blöcke folgen also bei der Ausgabe dem Kaltstartprogramm. Bei dieser Aufeinanderfolge der unterschiedlichen Kassettenverfahren wird der Vorteil von SuperTape deutlich vor Augen — und vor Ohren — geführt: Für das Kaltladeprogramm (circa ½ K) braucht der ZX 81 fast genauso lang wie für die Übertragung des gesamten Restes des 16K-Speichers mit SuperTape bei 7200 Bd (rund 17 s).

Während die File-Typen BAS und KAL keine weiteren Adreßangaben benötigen, müssen Sie beim Retten von Hexfiles: Typ HEX, sowohl die Start- als auch die Endadresse eingeben. Beim Laden können Sie eine Startadresse vorgeben, so daß der Datenblock in einen beliebigen Adreßbereich geladen werden kann. Auch beim BAS-Typ ist solch eine Vorgabe möglich, die den Standardwert übersteuert. Das ist sinnvoll, wenn zum Beispiel BASIC-Programme von anderen Rechnern geladen werden sollen, die die gleiche Typbezeichnung benutzen. Der ZX 81 braucht dann aber noch die entsprechenden Übersetzungsprogramme, um das Fremd-BASIC verstehen zu können.

In der Typcode-Tabelle ab Zeile 2040 des Listings sind noch einige Plätze zur freien Verfügung. Hier können Sie hinter den drei Bytes für den Typ die Adresse (lo,hi) eigener Unterprogramme einbringen, zum Beispiel für das Zusammenfügen von Programmen und/oder Daten, ASCII-Umformung des ganzen Datenblocks und so weiter.

Die Eingabe der Adressen kann — und sollte — im Hexcode erfolgen. Günstiger als ein angehängtes 'h' erschien dem Autor ein vorangehendes '$' — in Anlehnung an die Assembler der 6500-Serie — zur Kennzeichnung einer Hexadezimalzahl. Ohne dieses vorangestellte '$' interpretiert der Rechner die Adresse — wie üblich — dezimal.

Mehrere Files können unter gleichen Namen, aber eventuell verschiedenem Typ in dem Befehlsstring angesprochen werden, wobei der Schrägstrich '/' als Trennungszeichen dient. Beispiel:

```
'S,SUPER.BAS/
 HEX,$7A00,$7B00/
 HEX,$7B40,$7B50'
```

Die Jokersymbole dienen dazu, beim Laden entweder ein Zeichen ('?') oder den ganzen Rest ('*') eines Namens oder Typs zu überlesen.

Beispiel:

```
'L,*.*/*.*/*.*'
```

liest die nächsten drei Files, 'komme, was da wolle'.

Die Ladekontrolle

Kein Kontrollämpchen ist erforderlich, um anzuzeigen, ob

beim Laden alles mit rechten Dingen zugegangen ist. Die Routine LDSCO überwacht den Ladevorgang und gibt während der Ladepausen und beim Überlesen falscher Blöcke ihre Meldungen über den Bildschirm aus. Da der ZX 81 aber nicht gleichzeitig laden und Display-Funktionen verrichten kann, schaltet sie sofort wieder in den FAST-Mode, sobald sich am EAR-Eingang etwas rührt. Um allerdings falsche Blöcke überlesen zu können, muß sie vorher eine kleine Pause von einigen Millisekunden erkannt haben.

Bei einem falschen Block erscheint ein 'B' vor dem Namen und den restlichen Parametern. Die Routine sucht dann so lange weiter, bis sie entweder den richtigen Block gefunden hat, oder bis sie mit BREAK gestoppt wird. Ein Prüfsummenfehler wird mit 'F' markiert und ein Verify-Fehler mit '<' und daraufhin mit Meldung 'F' abgebrochen. Nach einem korrekten Ladevorgang bleibt das erste Feld leer und LDSCO gibt die Kontrolle an das aufrufende Hauptprogramm zurück, das so lange weiterläuft, bis der String abgearbeitet ist.

Ein 'A' hinter dem Typ signalisiert den ASCII-Code und ein 'Q' die Baudrate von 7200 für den Datenblock. Dann schließen sich die Startadresse und Blocklänge (hexadezimal!) an.

Grundsätzlich läßt sich jeder LOAD/SAVE-Vorgang mit BREAK abbrechen. Nur während des Kaltstartprogramms, solange das übliche Sinclair-Geräusch zu hören ist, führt ein BREAK normalerweise zum Absturz.

Ein-/Ausgabe-Utilities

Die folgenden Utilities lassen sich auch gut für eigene Maschinenprogramme nutzen:

RDHAD liest die durch CHADD angezeigte hexadezimale Adresse,

RDDAD macht das gleiche mit einer dezimal eingegebenen Adresse,

RDADR überprüft, ob die Eingabe hexadezimal (mit vorangehendem $) oder dezimal erfolgt ist,

PRCHA bringt ein Zeichen vom Akkumulator auf den Bildschirm, und zwar so, daß kein System-Crash daraus folgen kann,

PRHAD zeigt den Inhalt der durch HL adressierten Speicherzelle hexadezimal an.

STRAT schließlich setzt RAMTOP auf den durch NRAMT vorgegebenen Wert und verändert entsprechend den Stackpointer und ERR-SP. Die umständliche Prozedur, mittels NEW einen Adreßbereich zu schützen, löscht ja unglücklicherweise den ganzen sonstigen Speicherbereich.

Die Eingabe des Hexcodes erfolgt — wenn Sie keinen Assembler Ihr eigen nennen können — am besten wieder mit Hilfe des Mikroassemblers (siehe c't 4/84, Seite 23), der ja auch schon die SuperTape-Basisroutinen recht zügig einzutippen erlaubte. Um den gewählten Speicherbereich zu schützen, sollten Sie zuerst POKE 16389,122 und NEW eingeben.

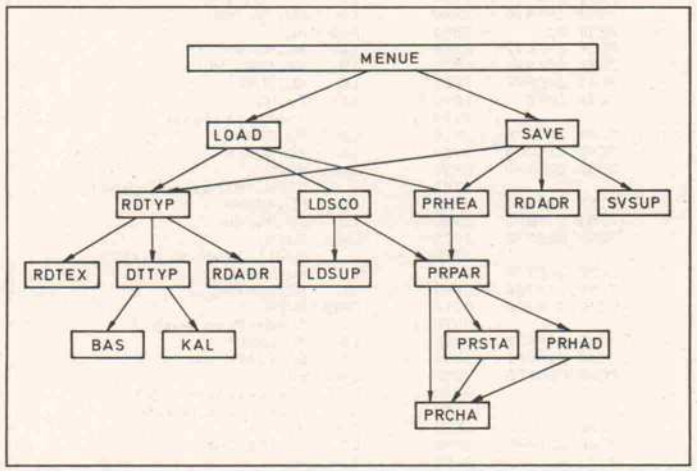


Bild 1 gibt einen Überblick über die Hierarchie der Ein-/Ausgaberoutinen.

```
                0010 ;          "**********"
7AB0            0020            ORG  7AB0H
7FCC            0030 ZFLAG      EQU  7FCCH
7FCD            0040 BDRAT      EQU  7FCDH
7FCE            0050 SVNAM      EQU  7FCEH
7FD9            0060 SVSL       EQU  7FD9H
7FDB            0070 SVTYP      EQU  7FDBH
7FDE            0080 SVFLA      EQU  7FDEH
7FDF            0090 SVSTA      EQU  7FDFH
7FE1            0100 SVLEN      EQU  7FE1H
7FE7            0110 LDNAM      EQU  7FE7H
7FF4            0120 LDTYP      EQU  7FF4H
7FF7            0130 LDFLA      EQU  7FF7H
7FF8            0140 LDSTA      EQU  7FF8H
7FFA            0150 LDLEN      EQU  7FFAH
7FFF            0160 LDEND      EQU  7FFFH
4021            0170 MODE       EQU  4021H
                0180 ;
7DA8            0190 SVSUP      EQU  7DA8H
7E54            0200 LDSUP      EQU  7E54H
7F73            0210 ZXASC      EQU  7F73H
7F83            0220 ASCZX      EQU  7F83H
                0230 ;
0207            0240 DAUER      EQU  0207H
02E7            0250 KFAST      EQU  02E7H
031E            0260 SVZXB      EQU  031EH
0F23            0270 FAST       EQU  0F23H
0F55            0280 EXPRN      EQU  0F55H
13F8            0290 STRAD      EQU  13F8H
4016            0300 CHADD      EQU  4016H
4014            0310 ELINE      EQU  4014H
4029            0320 NXTLN      EQU  4029H
4004            0330 RAMT       EQU  4004H
4002            0340 ERRSP      EQU  4002H
7900            0350 NRAMT      EQU  7900H
                0360 ;          "MENUE"
7AB0 CDE702     0370            CALL KFAST
7AB3 E7         0380            RST  20H
7AB4 CD550F     0390            CALL EXPRN
7AB7 CDF813     0400            CALL STRAD
7ABA ED531640   0410            LD   (CHADD),DE
7ABE DF         0420            RST  18H
7ABF 322140     0430            LD   (MODE),A
                0440 ;          "MODE=1.Zeichen"
7AC2 21CC7F     0450            LD   HL,ZFLAG
7AC5 3640       0460            LD   (HL),40H
7AC7 E7         0470 NXTCH      RST  20H
7AC8 21CC7F     0480            LD   HL,ZFLAG
7ACB FE3A       0490            CP   3AH
7ACD 2004       0500            JR   NZ,NOT-U
7ACF CBB6       0510            RES  6,(HL)
                0520 ;          "U: ASCII-Umwandl"
7AD1 18F4       0530            JR   NXTCH
7AD3 FE33       0540 NOT-U      CP   33H
7AD5 2004       0550            JR   NZ,NOT-N
                0560 ;          "N: kein Display"
7AD7 CBDE       0570            SET  3,(HL)
7AD9 18EC       0580            JR   NXTCH
7ADB FE1A       0590 NOT-N      CP   1AH
7ADD 2024       0600            JR   NZ,ERVAL
                0610 ;          "komma 1.trennz."
7ADF 060C       0620            LD   B,0CH
7AE1 11CE7F     0630            LD   DE,SVNAM
7AE4 CD9B7C     0640            CALL RDTEX
                0650 ;          "bestimmt Name"
7AE7 DF         0660            RST  18H
7AE8 FE1B       0670            CP   1BH
                0680 ;          "Punkt 2. Trennz."
7AEA 2017       0690            JR   NZ,ERVAL
7AEC 12         0700            LD   (DE),A
7AED 3A2140     0710 NXTYP      LD   A,(MODE)
7AF0 21CC7F     0720            LD   HL,ZFLAG
                0730 ;          "Sprung zu den"
                0740 ;          "Behandlungsrout"
7AF3 FE31       0750            CP   31H
7AF5 2810       0760            JR   Z,LOAD
7AF7 FE3B       0770            CP   3BH
7AF9 280A       0780            JR   Z,VERI
7AFB FE36       0790            CP   36H
7AFD 2818       0800            JR   Z,QUICK
7AFF FE38       0810            CP   38H
7B01 2816       0820            JR   Z,SAVE
7B03 CF         0830 ERVAL      RST  8
7B04 0B         0840            DEFB 0BH
7B05 CBCE       0850 VERI       SET  1,(HL)
7B07 CD5A7B     0860 LOAD       CALL RDTYP
                0870 ;          "best. Typ"
7B0A CDBE7C     0880            CALL PRHEA
                0890 ;          "Param.-Ausgabe"
7B0D CD3C7D     0900            CALL LDSCO
                0910 ;          "Ladekontrolle"
7B10 DF         0920 CKEND      RST  18H
7B11 FE18       0930            CP   18H
7B13 28D8       0940            JR   Z,NXTYP
                0950 ;          "bei / weiter"
7B15 CF         0960            RST  8
7B16 FF         0970            DEFB 0FFH
7B17 CBFE       0980 QUICK      SET  7,(HL)
                0990 ;          "Flag fuer 7200 Bd"
7B19 CBC6       1000 SAVE       SET  0,(HL)
7B1B CD5A7B     1010            CALL RDTYP
                1020 ;          "best. Typ+Startadr."
7B1E DF         1030            RST  18H
7B1F FE1A       1040            CP   1AH
```

```
7B21 200A      1050        JR   NZ,SKADR
               1060 ;           "Komma + Endadr."
7B23 E7        1070        RST  20H
7B24 CD777C    1080        CALL RDADR
               1090 ;           "decod. Adresse"
7B27 2ADF7F    1100        LD   HL,(SVSTA)
7B2A CDC27B    1110        CALL DTLEN
7B2D 21CC7F    1120 SKADR  LD   HL,ZFLAG
7B30 CB76      1130        BIT  6,(HL)
7B32 200D      1140        JR   NZ,NOASC
               1150 ;           "ASCII - Umwandl."
7B34 11CE7F    1160        LD   DE,SVNAM
7B37 0610      1170        LD   B,10H
7B39 1A        1180 SVASC  LD   A,(DE)
7B3A CD737F    1190        CALL ZXASC
7B3D 12        1200        LD   (DE),A
7B3E 13        1210        INC  DE
7B3F 10F8      1220        DJNZ SVASC
7B41 CB66      1230 NOASC  BIT  4,(HL)
               1240 ;           "=1 fuer Kaltstart"
7B43 CBA6      1250        RES  4,(HL)
7B45 C42E7C    1260        CALL NZ,SVKAL
7B48 010080    1270        LD   BC,8000H
7B4B 0B        1280 PAUSE  DEC  BC
7B4C 78        1290        LD   A,B
7B4D B1        1300        OR   C
7B4E 20FB      1310        JR   NZ,PAUSE
7B50 1603      1320        LD   D,03H
7B52 CDBE7C    1330        CALL PRHEA
               1340 ;           "Param.-ausgabe"
7B55 CDA87D    1350        CALL SVSUP
               1360 ;           "Supersave"
7B58 18B6      1370        JR   CKEND
               1380 ;           "Test auf Ende"
               1390 ;           "**************"
               1400 ;           "liest Typ"
7B5A 0603      1410 RDTYP  LD   B,03H
7B5C 11DB7F    1420        LD   DE,SVTYP
7B5F CD9B7C    1430        CALL RDTEX
7B62 CD7B7B    1440        CALL DTTYP
7B65 1600      1450        LD   D,00
7B67 DF        1460        RST  18H
7B68 FE1A      1470        CP   1AH
7B6A C0        1480        RET  NZ
               1490 ;           "Komma: lese"
               1500 ;           "Startadresse"
7B6B E7        1510        RST  20H
7B6C CD777C    1520        CALL RDADR
7B6F ED53DF7F  1530        LD   (SVSTA),DE
7B73 21CC7F    1540        LD   HL,ZFLAG
7B76 CBD6      1550        SET  2,(HL)
7B78 1601      1560        LD   D,01
7B7A C9        1570        RET
               1580 ;           "************"
               1590 ;           "best. Adresse"
               1600 ;           "fuer Typ"
7B7B 0E06      1610 DTTYP  LD   C,06H
7B7D 11DB7F    1620        LD   DE,SVTYP
7B80 219D7B    1630        LD   HL,TYTAB
7B83 D5        1640 NXTTW  PUSH DE
7B84 0603      1650        LD   B,03H
7B86 1A        1660 NXTTC  LD   A,(DE)
7B87 BE        1670        CP   (HL)
               1680 ;           "Zeichen in Tabelle?"
7B88 23        1690        INC  HL
               1700 ;           "nein: naechster Typ"
7B89 2009      1710        JR   NZ,UNGL
               1720 ;           "ja: teste naechstes"
               1730 ;           "Zeichen"
7B8B 13        1740        INC  DE
7B8C 10F8      1750        DJNZ NXTTC
               1760 ;           "Typ erkannt"
7B8E D1        1770        POP  DE
7B8F 5E        1780        LD   E,(HL)
7B90 23        1790        INC  HL
7B91 56        1800        LD   D,(HL)
7B92 EB        1810        EX   HL,DE
               1820 ;           "HL:= Behandl.-Adr."
7B93 E9        1830        JP   (HL)
7B94 23        1840 UNGL   INC  HL
7B95 10FD      1850        DJNZ UNGL
7B97 23        1860        INC  HL
7B98 0D        1870        DEC  C
7B99 D1        1880        POP  DE
7B9A 20E7      1890        JR   NZ,NXTTW
               1900 ;           "nicht gefunden"
7B9C C9        1910        RET
               1920 ;           "************"
               1930 ;           "Typtabelle"
7B9D 27        1940 TYTAB  DEFB 27H
7B9E 26        1950        DEFB 26H
7B9F 38        1960        DEFB 38H
7BA0 BB7B      1970        DEFW BAS
7BA2 30        1980        DEFB 30H
7BA3 26        1990        DEFB 26H
7BA4 31        2000        DEFB 31H
7BA5 CD7B      2010        DEFW KAL
               2020 ;           "Platz fuer weitere"
               2030 ;           "Typen"
0014           2040        DEFS 20
               2050 ;           "************"
               2060 ;           "Typ: BAS"
7BBB 210940    2070 BAS    LD   HL,4009H
```

```
7BBE ED5B1440  2080        LD   DE,(ELINE)
               2090 ;           "**********"
               2100 ;           "best. Startadr"
               2110 ;           "und Laenge"
7BC2 22DF7F    2120 DTLEN  LD   (SVSTA),HL
7BC5 EB        2130        EX   HL,DE
7BC6 A7        2140        AND  A
7BC7 ED52      2150        SBC  HL,DE
7BC9 22E17F    2160        LD   (SVLEN),HL
7BCC C9        2170        RET
               2180 ;           "Typ: KAL"
7BCD 210079    2190 KAL    LD   HL,NRAMT
7BD0 11547E    2200        LD   DE,LDSUP
7BD3 CDC27B    2210        CALL DTLEN
               2220 ;           "setzt Param. fuer"
               2230 ;           "Masch.-Progs ab NRAMT"
7BD6 21CC7F    2240        LD   HL,ZFLAG
7BD9 CBE6      2250        SET  4,(HL)
               2260 ;           "Kaltst.-Flag =1"
7BDB 21E27B    2270        LD   HL,CHBAS
7BDE 221640    2280        LD   (CHADD),HL
               2290 ;           "danach Typ:BAS"
7BE1 C9        2300        RET
7BE2 18        2310 CHBAS  DEFB 18H
7BE3 27        2320        DEFB 27H
7BE4 26        2330        DEFB 26H
7BE5 38        2340        DEFB 38H
7BE6 0B        2350        DEFB 0BH
               2360 ;           "**********"
               2370 ;           "ausgesend. BASIC"
               2380 ;           "bei Kaltstart"
7BE7 0001      2390 LDKAL  DEFW 0100H
7BE9 0E00      2400        DEFW 000EH
7BEB F5D4      2410        DEFW 0D4F5H
7BED 1D22      2420        DEFW 221DH
7BEF 2120      2430        DEFW 2021H
7BF1 257E      2440        DEFW 7E25H
7BF3 8F01      2450        DEFW 018FH
7BF5 4A00      2460        DEFW 004AH
7BF7 0076      2470        DEFW 7600H
               2480 ;           "setzt neues RAMTOP"
7BF9 D1        2490 STRAT  POP  DE
7BFA 210079    2500        LD   HL,NRAMT
7BFD 220440    2510        LD   (RAMT),HL
7C00 F9        2520        LD   SP,HL
7C01 21003E    2530        LD   HL,3E00H
7C04 E5        2540        PUSH HL
7C05 217606    2550        LD   HL,0676H
7C08 E5        2560        PUSH HL
7C09 ED730240  2570        LD   (ERRSP),SP
7C0D D5        2580        PUSH DE
7C0E C9        2590        RET
               2600 ;           "verschiebt LDSUP"
7C0F CD8F40    2610        CALL 408FH
7C12 21C440    2620        LD   HL,40C4H
7C15 11547E    2630        LD   DE,LDSUP
7C18 01AC01    2640        LD   BC,01ACH
7C1B EDB0      2650        LDIR
7C1D CD547E    2660        CALL LDSUP
               2670 ;           "laedt Masch.-Progs"
7C20 3E31      2680        LD   A,31H
7C22 322140    2690        LD   (MODE),A
7C25 21E27B    2700        LD   HL,CHBAS
7C28 221640    2710        LD   (CHADD),HL
               2720 ;           "laedt BASIC"
7C2B C3ED7A    2730        JP   NXTYP
               2740 ;           "**********"
               2750 ;           "Save Superload"
               2760 ;           "im ZX- Verfahren"
7C2E CD230F    2770 SVKAL  CALL FAST
               2780 ;           "ELINE und NXTLN"
               2790 ;           "werden neu gesetzt"
7C31 2A2940    2800        LD   HL,(NXTLN)
7C34 E5        2810        PUSH HL
7C35 217D40    2820        LD   HL,407DH
7C38 222940    2830        LD   (NXTLN),HL
7C3B 2A1440    2840        LD   HL,(ELINE)
7C3E E5        2850        PUSH HL
7C3F 217042    2860        LD   HL,4270H
7C42 221440    2870        LD   (ELINE),HL
7C45 21D97F    2880        LD   HL,SVSL
7C48 CBFE      2890        SET  7,(HL)
               2900 ;           "Namensausgabe"
7C4A 21CE7F    2910        LD   HL,SVNAM
7C4D 010C00    2920        LD   BC,000CH
7C50 CD877D    2930        CALL SVZX
               2940 ;           "Sys.-Param.-Ausgabe"
7C53 210940    2950        LD   HL,4009H
7C56 017400    2960        LD   BC,0074H
7C59 CD877D    2970        CALL SVZX
               2980 ;           "Kaltst.-Prog Ausg."
7C5C 21E77B    2990        LD   HL,LDKAL
7C5F 014700    3000        LD   BC,0047H
7C62 CD877D    3010        CALL SVZX
               3020 ;           "Lade-Prog Ausg."
7C65 21547E    3030        LD   HL,LDSUP
7C68 01AE01    3040        LD   BC,01AEH
7C6B CD877D    3050        CALL SVZX
               3060 ;           "Ruecksetzen von"
               3070 ;           "ELINE und NXTLN"
7C6E E1        3080        POP  HL
7C6F 221440    3090        LD   (ELINE),HL
7C72 E1        3100        POP  HL
```

```
7C73 222940    3110        LD   (NXTLN),HL
7C76 C9        3120        RET
               3130 ;           "**********"
               3140 ;           "liest Adresse"
7C77 FE0D      3150 RDADR  CP   0DH
7C79 2809      3160        JR   Z,RDHAD
7C7B CDD914    3170 RDDAD  CALL 14D9H
7C7E CDA70E    3180        CALL 0EA7H
7C81 50        3190        LD   D,B
7C82 59        3200        LD   E,C
7C83 C9        3210        RET
               3220 ;           "**********"
7C84 E7        3230 RDHAD  RST  20H
7C85 110000    3240        LD   DE,0000H
7C88 42        3250        LD   B,D
7C89 D61C      3260 NXTZI  SUB  1CH
7C8B FE10      3270        CP   10H
               3280 ;           "Ret falls <>Hexzif."
7C8D D0        3290        RET  NC
7C8E EB        3300        EX   HL,DE
7C8F 29        3310        ADD  HL,HL
7C90 29        3320        ADD  HL,HL
7C91 29        3330        ADD  HL,HL
7C92 29        3340        ADD  HL,HL
7C93 4F        3350        LD   C,A
7C94 09        3360        ADD  HL,BC
7C95 EB        3370        EX   HL,DE
7C96 CD4900    3380        CALL 0049H
7C99 18EE      3390        JR   NXTZI
               3400 ;           "**********"
               3410 ;           "liest Name/Typ"
7C9B CD4900    3420 RDTEX  CALL 0049H
7C9E 04        3430        INC  B
7C9F FE18      3440        CP   18H
7CA1 3804      3450        JR   C,SOND
7CA3 FE1C      3460        CP   1CH
7CA5 3810      3470        JR   C,CLEAR
7CA7 FE0B      3480 SOND   CP   0BH
7CA9 280C      3490        JR   Z,CLEAR
7CAB FE40      3500        CP   40H
7CAD 3008      3510        JR   NC,CLEAR
7CAF 05        3520        DEC  B
7CB0 28E9      3530        JR   Z,RDTEX
7CB2 12        3540        LD   (DE),A
7CB3 13        3550        INC  DE
7CB4 05        3560        DEC  B
7CB5 18E4      3570        JR   RDTEX
7CB7 05        3580 CLEAR  DEC  B
               3590 ;           "fuellt mit Spaces auf"
7CB8 C8        3600        RET  Z
7CB9 AF        3610        XOR  A
7CBA 12        3620        LD   (DE),A
7CBB 13        3630        INC  DE
7CBC 18F9      3640        JR   CLEAR
               3650 ;           "**********"
               3660 ;           "ueberschrift"
7CBE 3ACC7F    3670 PRHEA  LD   A,(ZFLAG)
7CC1 5F        3680        LD   E,A
7CC2 E6C0      3690        AND  0C0H
7CC4 32DE7F    3700        LD   (SVFLA),A
7CC7 AF        3710        XOR  A
7CC8 32CD7F    3720        LD   (BDRAT),A
7CCB CB5B      3730        BIT  3,E
7CCD C0        3740        RET  NZ
7CCE 3A2140    3750        LD   A,(MODE)
7CD1 21CE7F    3760        LD   HL,SVNAM
               3770 ;           "**********"
               3780 ;           "Param. Display"
7CD4 CD0E7D    3790 PRPAR  CALL PRCHA
               3800 ;           "print Mode/Fehler"
7CD7 AF        3810        XOR  A
7CD8 D7        3820        RST  10H
7CD9 0610      3830        LD   B,10H
               3840 ;           "Print Name/Typ"
7CDB 7E        3850 PRNCH  LD   A,(HL)
7CDC CB73      3860        BIT  6,E
7CDE CC837F    3870        CALL Z,ASCZX
7CE1 CD0E7D    3880        CALL PRCHA
7CE4 23        3890        INC  HL
7CE5 10F4      3900        DJNZ PRNCH
7CE7 AF        3910        XOR  A
7CE8 D7        3920        RST  10H
7CE9 CB76      3930        BIT  6,(HL)
7CEB 2002      3940        JR   NZ,NOA
7CED 3E26      3950        LD   A,26H
               3960 ;           "ASCII ? Print A"
7CEF D7        3970 NOA    RST  10H
7CF0 AF        3980        XOR  A
7CF1 D7        3990        RST  10H
7CF2 CB7E      4000        BIT  7,(HL)
7CF4 2802      4010        JR   Z,NO-Q
7CF6 3E36      4020        LD   A,36H
               4030 ;           "Quick ? Print Q"
7CF8 D7        4040 NO-Q   RST  10H
7CF9 23        4050        INC  HL
7CFA CB42      4060        BIT  0,D
               4070 ;           "Print Adr. bzw. Sterne"
7CFC CC317D    4080        CALL Z,PRSTA
7CFF C4167D    4090        CALL NZ,PRHAD
7D02 CB4A      4100        BIT  1,D
7D04 CC317D    4110        CALL Z,PRSTA
7D07 C4167D    4120        CALL NZ,PRHAD
7D0A 3E76      4130        LD   A,76H
```

```
7D0C D7        4140        RST  10H
               4150 ;           "Newline"
7D0D C9        4160        RET
               4170 ;           "***********"
               4180 ;           "print Zeichen"
7D0E FE76      4190 PRCHA  CP   76H
7D10 2802      4200        JR   Z,NO-NL
7D12 E6BF      4210        AND  0BFH
7D14 D7        4220 NO-NL  RST  10H
7D15 C9        4230        RET
               4240 ;           "***********"
               4250 ;           "print Hexadresse"
7D16 AF        4260 PRHAD  XOR  A
7D17 D7        4270        RST  10H
7D18 7E        4280        LD   A,(HL)
7D19 F5        4290        PUSH AF
7D1A 23        4300        INC  HL
7D1B 7E        4310        LD   A,(HL)
7D1C CD217D    4320        CALL PRBYT
7D1F F1        4330        POP  AF
7D20 23        4340        INC  HL
               4350 ;           "**********"
               4360 ;           "print Byte"
7D21 F5        4370 PRBYT  PUSH AF
7D22 E6F0      4380        AND  0F0H
7D24 1F        4390        RRA
7D25 1F        4400        RRA
7D26 1F        4410        RRA
7D27 1F        4420        RRA
7D28 CDEB07    4430        CALL 07EBH
7D2B F1        4440        POP  AF
7D2C E60F      4450        AND  0FH
7D2E C3EB07    4460        JP   07EBH
               4470 ;           "print Sterne"
7D31 AF        4480 PRSTA  XOR  A
7D32 D7        4490        RST  10H
7D33 0604      4500        LD   B,04H
7D35 3E17      4510 NXTST  LD   A,17H
7D37 D7        4520        RST  10H
7D38 10FB      4530        DJNZ NXTST
7D3A AF        4540        XOR  A
7D3B C9        4550        RET
               4560 ;           "***********"
               4570 ;           "ladeKontrolle"
7D3C AF        4580 LDSCO  XOR  A
7D3D 32CD7F    4590        LD   (BDRAT),A
7D40 CD0702    4600        CALL DAUER
               4610 ;           "Warte auf Pause"
7D43 0600      4620 WTPAU  LD   B,00H
7D45 3E7F      4630 WTCON  LD   A,7FH
7D47 DBFE      4640        IN   A,(0FEH)
7D49 17        4650        RLA
7D4A 38F7      4660        JR   C,WTPAU
7D4C 1F        4670        RRA
7D4D 1F        4680        RRA
7D4E 3035      4690        JR   NC,BREAK
7D50 10F3      4700        DJNZ WTCON
               4710 ;           "Warte auf Puls"
7D52 3E7F      4720 WTPUL  LD   A,7FH
7D54 DBFE      4730        IN   A,(0FEH)
7D56 FE7F      4740        CP   7FH
7D58 28F8      4750        JR   Z,WTPUL
7D5A 57        4760        LD   D,A
7D5B 17        4770        RLA
7D5C 3027      4780        JR   NC,BREAK
7D5E CDE702    4790        CALL KFAST
7D61 CD547E    4800        CALL LDSUP
7D64 FD210040  4810        LD   IY,4000H
7D68 21CC7F    4820        LD   HL,ZFLAG
7D6B CB5E      4830        BIT  3,(HL)
7D6D 200E      4840        JR   NZ,NXTBL
7D6F ED5BF77F  4850        LD   DE,(LDFLA)
               4860 ;           "E :=gelad.Flags"
7D73 F5        4870        PUSH AF
7D74 21E77F    4880        LD   HL,LDNAM
7D77 1603      4890        LD   D,03
7D79 CDD47C    4900        CALL PRPAR
7D7C F1        4910        POP  AF
7D7D FE27      4920 NXTBL  CP   27H
               4930 ;           "falscher Block ?"
               4940 ;           "weiter"
7D7F 28BB      4950        JR   Z,LDSCO
7D81 A7        4960        AND  A
               4970 ;           "RET falls fehlerfrei"
7D82 C8        4980        RET  Z
               4990 ;           "sonst Abbruch"
7D83 CF        5000        RST  8
7D84 0E        5010        DEFB 0EH
               5020 ;           "**********"
7D85 CF        5030 BREAK  RST  08
7D86 0C        5040        DEFB 0CH
               5050 ;           "**********"
               5060 ;           "Save im ZX Verf."
               5070 ;           "HL: Start, BC: Laenge"
7D87 E5        5080 SVZX   PUSH HL
7D88 C5        5090        PUSH BC
7D89 CD1E03    5100        CALL SVZXB
7D8C C1        5110        POP  BC
7D8D E1        5120        POP  HL
7D8E EDA1      5130        CPI
7D90 E0        5140        RET  PO
7D91 18F4      5150        JR   SVZX
               5160        END
```

c't 86 — Zwischenbilanz

Eberhard Meyer, Detlef Grell

Die Resonanz, die der c't 86 bei unserer Leserschaft fand, hat unsere Erwartungen bei weitem übertroffen. Am meisten verblüffte uns, daß viele Leser, die sich bisher der Faszination des Mikrocomputers erfolgreich entzogen hatten, ausgerechnet mit einem 16-Bit-System den Einstieg wagen wollen. Etliche versuchen sich dabei sogar zum erstenmal im Computerselbstbau. Darauf waren die bisherigen Beiträge zum c't 86 allerdings nicht abgestimmt. So werden an dieser Stelle ein paar grundsätzliche Aspekte des Nachbaus erläutert, und nicht zuletzt muß auch auf einige Fehler (keine Angst, nur ganz wenige sind 'funktionshemmend') in den bisherigen Beschreibungen hingewiesen werden.

Es wurde zwar in den Beiträgen zum c't 86 bereits mehrfach erwähnt, es ist aber wohl nicht recht geglaubt worden:

Der Selbstbau des c't 86 ist kein Projekt für Anfänger!

Wer nicht wenigstens gut löten kann, hat buchstäblich 'schlechte Karten'. Ein kleines Experiment der Redaktion, die einen Satz Platinen in 'Heimarbeit' bestücken ließ, erbrachte ebendies: Vier schlechte Karten, keine lief auf Anhieb! Wer zudem nicht über ein grundsätzliches Verständnis der Funktionsweise von Mikroprozessorsystemen verfügt (oder nicht zumindest Freunde hat, die ihn tatkräftig unterstützen können), wird seine liebe Not bei der Fehlersuche haben. Und letztere wird mit großer Wahrscheinlichkeit erforderlich sein. Das hat nichts mit 'geringer Nachbausicherheit' zu tun. Ein defektes Bauteil, ein IC, das seine Spezifikation nicht einhält (was erheblich öfter vorkommt, als man gemeinhin glaubt), und guter Rat wird in der Tat teuer.

Wer mit Begriffen wie 'RAS', 'Segment Register', 'Head load' oder 'Interrupt' nur vage Vorstellungen verbindet, sollte doch ernsthaft den Kauf von Fertigkarten erwägen. Dabei geht es uns keineswegs darum, Bausatzhändlern in die Hände zu arbeiten. Aber der Leser erspart sich und letztlich auch uns eine Menge Frust; denn nur zu schnell sind enttäuschte Nachbauer bereit, die Schuld ausschließlich bei den 'Urhebern' der Schaltung zu suchen.

Die ausgearbeitete Methode zur Inbetriebnahme geht zwar erheblich über das hinaus, was man vielfach in Bauanleitungen geboten bekommt, dennoch wird die — zugegeben — knappe Beschreibung und Dokumentation beim Einsteiger viele Fragen unbeantwortet lassen. Fairerweise muß man allerdings sagen, daß das Fehlen der Signalbezeichnungen an ICs und Steckerleisten auch den 'Profis' einige Mühe machte. Wir geloben Besserung.

Korrekturen

Wie bei einem Projekt dieser Größenordnung nicht anders zu erwarten, haben sich trotz größter Sorgfalt einige Fehler eingeschlichen. Nach Abschluß der Entwicklung im September 83, als unsere sehr 'prototyphaften' Platinen endlich fehlerfrei liefen, ging es zum CAD-Layouten. Dazu mußten aus dem Schaltplan Verbindungslisten erstellt werden. Wie nicht anders zu erwarten, liefen sämtliche Erstlayouts nicht. Nach den Korrekturen wurden dann die Bestückungspläne gezeichnet. Nun ja, keine der Entwicklungsstufen hat ihre Chance, Fehler ins Projekt hineinzubringen, ungenutzt gelassen.

Aber keine Sorge, das meiste erschöpft sich in kleinen Ungereimtheiten, die sich anhand von Schaltplan, Text, Bestückungsplan und fertiger Platine von selbst klären. Trotzdem führen wir sie hier der Vollständigkeit halber mit auf, die wirklich gravierenden Fehler jeweils zuerst.

CPU-Karte

Auf einigen Platinen fehlt eine Leiterbahn zwischen IC4/Pin 4 und der Steckerleiste X1C/24. Mit einem Ohmmeter kann schnell überprüft werden, ob diese Verbindung mit einem Draht hergestellt werden muß.

R1 und R2 betragen 1,5 kOhm, nicht 150 Ohm.

C1 beträgt nicht 10µF, wie in der Stückliste aufgeführt, sondern 100µF/10 V.

C2 ist ebenfalls in der Stückliste falsch angegeben. Korrekter Wert: 100pF.

Die Kondensatoren C12, C13 und C14 sind im Bestückungsaufdruck als Elektrolytkondensatoren eingezeichnet. Hier können wahlweise keramische Kondensatoren mit 100 nF oder induktionsarme Tantal-Kondensatoren 10µF/16 V eingesetzt werden. Obwohl erstere meistens genügen, läßt sich die Betriebssicherheit mit den Tantal-Typen erhöhen.

Im Schaltbild hat IC22 (Pin 1,2,3) einen Negationspunkt 'abbekommen', obwohl es ein ODER-Gatter ist.

Pin 31 von IC1 ist mit Pin 26 der Steckerleiste X1B verbunden, nicht mit Pin 25.

X1C/24 ist nicht mit Pin 12 von IC19 verbunden, sondern mit IC11/Pin 1.

Bei der Brücke BR1 sind, wie im Schalt- und Bestückungsplan angegeben, Pin 2 und 3 miteinander zu verbinden.

In der Regel kann man die Pfostenleiste für BR3 einsparen und die Lötaugen direkt miteinander verbinden. Dadurch wird die Abfrage der beiden externen (Bus-)Waitleitungen aktiviert.

BR4 (für zusätzliche Interrupts) und BR5/6 (für Betrieb mit dem Arithmetik-Prozessor) sind in der derzeitigen Ausbaustufe uninteressant und werden nicht gesetzt.

Die Tabellen zum Setzen von BR7, 9, 10, 11, 12 sind im Schaltplan zu finden. In der unteren Tabelle muß es BR10, nicht BR11 heißen. 'Kürzere' EPROMs werden so eingesetzt, daß ihre Pins 1 und 24 in den Pins 3 und 26 der IC-Fassung stecken.

RAM-Karte

IC15 ist in Schaltbild und Stückliste als LS241 angegeben worden. Hier **muß** ein **LS244** eingesetzt werden.

Auf Bestückungsplan und -aufdruck ist die Lage von C14 falsch eingezeichnet. Wenn der Kondensator so eingebaut wird, dann liegt er zwischen Masse und einer Signalleitung. Er muß jedoch zwischen Masse und Plus (freies Lötauge neben C34) eingesetzt werden.

Im Bestückungsplan (und -aufdruck) ist C1 nicht polarisiert angegeben. Der 10µF Tantalkondensator ist mit dem positiven Anschluß zur Steckerleiste X1 hin einzulöten.

IC12 ist in der Stückliste und im Schaltplan als 74S04 angegeben, eine normale LS-Version ist ausreichend.

Als RAM-Baustein können alle handelsüblichen 64Kx1-Chips mit maximal 150ns Zugriffszeit verwendet werden, also zum Beispiel 4164, 4864 oder i2164. Zu beachten ist jedoch, daß es ICs gibt, die mit 256 Refresh-Adressen arbeiten. Im Zweifelsfalle sollte man das Datenblatt der verwendeten ICs zu Rate ziehen. Beide Versionen (128er und 256er Refresh) sind verwendbar, es muß lediglich die Brücke BR1 entsprechend gesetzt werden.

Es gibt Hersteller, die für die Betriebsspannung der RAM-ICs eine Kapazität von 100µF verlangen. Dem kann durch Vergrößern von C34 bis 37 auf 22µF Rechnung getragen werden.

I/0-Karte

Einigen Platinen fehlt die Verbindung zwischen IC4/Pin 4 und IC5/Pin 9. Sie ist gegebenenfalls mit einem Draht herzustellen.

Für das Widerstandsnetzwerk RN1 ist eines mit 7 Widerständen ausreichend. Da Arrays mit 8 Widerständen aber leichter beschaffbar sind, wurde ein zusätzliches Loch auf der Platine vorgesehen. Das Lötloch für Pin 9 ist also ohne Funktion.

C15 fehlt in der Stückliste. Der Wert beträgt 47pF. Am Stecker X5 ist Pin 1 'TTL-out', Pin 3 'TTL-in'.

c't-Einzelheft-Bestellung

c't können Sie ab Ausgabe 12/83 direkt beim Verlag nachbestellen. Den Betrag von DM 6,— (zuzügl. DM 2,— für Porto und Verpackung) überweisen Sie bitte auf eines unserer Konten* oder fügen Sie Ihrer Bestellung einen Verrechnungsscheck bei.

Bei Bestellungen bis zu 4 Heften erhöhen sich die Kosten für Porto und Verpackung auf DM 3,—.

* s. Seite 111

ELEKTRONIKA SAAR

7. - 9. SEPT. 1984

KONGRESSHALLE SAARBRÜCKEN

VERKAUFS – UND INFORMATIONSMESSE
FÜR ELEKTRONIK UND HEIMCOMPUTER

FÜR DEN HOBBY- UND PROFIBEREICH
FÜR BASTLER UND ANWENDER .....

INFORMATION ....
ELEKTRONIKA SAAR
POSTFACH 10 12 60
6620 VÖLKLINGEN

NEWMAN Computer~Versand
Rolf W. Neumann

Alles für den Home-Computer

NEWMAN BERATUNGS-KATALOG

Gleich anfordern!
Kostenlos und unverbindlich erhalten Sie den großen Home-Computer Beratungs-Katalog mit vielen tollen Angeboten. Einfach Coupon einsenden.

GRATIS

Wir haben nur Qualitäts-Markenartikel zu Preisen, die uns so leicht keiner nachmacht.
Wir sind ein Versandhaus nur für Home-Computer, Hardware, Programme und Zubehör.
Wir beraten Sie neutral und unverbindlich. Am besten rufen Sie uns an. Wir helfen Ihnen weiter.
Wir liefern in der Regel innerhalb von 8 Tagen. Sofort-Lieferbestätigung bei telefonischer Anfrage.
Und außerdem erhalten Sie die Original-Hersteller-Garantie auf alle Artikel.
Teilzahlung ab sofort auch möglich.

Commodore 64
Newman liefert alles sofort, wer kann Ihnen das sonst noch bieten! Die gesamte Peripherie original von COMMODORE sofort ab Lager lieferbar. Und das zu Preisen, die uns so leicht keiner nachmacht. Fragen Sie unbedingt nach unseren Paket-Angeboten. **Sie werden staunen.**

Neu! Commodore SX 64
tragbarer "C 64" 64 K, 170 KB, – Floppy und Farb-Monitor zum Sonder-Preis von DM **2.948,–**
(kein Druckfehler)

Sharp MZ – 721
64 K, integrierter Cassetten-Recorder, 10 Spielprogramme gratis **nur** DM **939,–**
(da nur begrenzte Stückzahl sofort lieferbar, bitte noch heute bestellen).

Sharp MZ – 731
wie MZ – 721, zusätzlich mit integriertem Vierfarben-Drucker, 20 Programme am Lager. **nur** DM **1.239,–**

Sinclair ZX Spectrum, 16 KB, RAM DM **394,90**

Sinclair Spectrum, 48 KB, RAM nur noch DM **539,–**

Dragon 32
32 K-RAM Super-Graphic DM **795,–**

Seikosha GP 100 A,
Matrix-Drucker, 50 Zeichen/sec. DM **675,–**

Sanyo, Daten-Monitor, 2112, gestochen scharf, grüne Anzeige DM **299,–**

Anzeige in orange-farben, 2212 DM **309,–**

Spectravideo, SV 318 DM **888,–**

Spectravideo, SV 328 DM **1.248,–**
(die gesamte Peripherie ist auch lieferbar).

Epson-Drucker
besonders preiswert, z. B. RX 80, nur DM **1.148,–**
Außerdem lieferbar: Texas Instruments, Brothers und jede Menge Fachbücher, Spiel- und Lern-Programme, Drucker, Laufwerke, Monitore und und und ...

Am besten gleich nachfragen: **040/830 26 27**
040/830 28 29

Ausschneiden auf Postkarte kleben (60 Pf Porto)

Ja, ☐ bitte senden Sie mir sofort kostenlos und unverbindlich Ihren Beratungs-Katalog. KT 04

Für Ihre Bestellung bitte hier eintragen. Alle Preise incl. MWSt. zuz. Versand-Kosten. Lieferung per Nachnahme, **Teilzahlung ab sofort möglich!**

Artikel	Stück	Preis

Name/Vorname
Straße
PLZ/Ort
Vorwahl/Telefon-Nr.
Unterschrift
Alter:

NEWMAN Computer~Versand
Rolf W. Neumann, Postfach 57 12 61, 2000 Schenefeld.

Personalcomputer & 100 MHz Logikanalysator revolutionieren die Meßtechnik

Der Computer ein IBM®-PC. Mit ihm verbunden ein Northwest Instrument's µAnalyst tm 2000.

Das Drücken einer Taste aktiviert die vereinten Möglichkeiten zu einem „State of the Art" Logik Analysator. Übrigens muß der Rechner kein IBM-PC sein. Der Columbia PC ist ebenfalls mit dem µAnalyst kompatibel.

Max. Abtastfrequenz	100 MHz (Zeit)
Max. Kanalzahl	80 State – 16 Timing
Max. Speichertiefe	4096 State
State Trigger	15 Ebenen IF-Then/Else-IF/Trace Control
Timing Trigger	14 Möglichkeiten, wie z. B. Transitional Trigger, Multi-Trigger und set up-Hold time-Trigger

Denken Sie über beides – Zeit und Zustands-Analyse, sowie über die Post-Datenverarbeitung nach.

Northwest Instrument unterstützt Disassembler für folgende Microprozessoren:

Z 80; 8080; 8085; 8086; 8088; 6800; 6802; 6808; 6301; 6502; 68000; 68008; 68010.

Rufen Sie uns einfach an und lassen Sie sich nicht nur von der Technik, sondern auch vom Preis überraschen.

Instrumatic® Electronic GmbH
Ihr Partner für Meßtechnik
Am Kirchenhölzl 14 · 8032 Gräfelfing
Tel. 089/85 20 63 · Telex 5 24 298 instr d

Technische Büros:
Stuttgart, Tel. (0 70 23) 44 20
Frankfurt, Tel. (0 61 82) 2 77 22
Düsseldorf, Tel. (02 03) 7 43 88
Hamburg, Tel. (0 41 83) 65 25
Berlin, Gärtner Electronic, Tel. (0 30) 8 52 90 21

NORTHWEST INSTRUMENT SYSTEMS, INC. NW iS

Floppy-Karte

Im Schaltbild liegt bei BR1 (Motor on) angeblich Pin 3 auf Masse. Richtig (auf der Platine) ist aber Pin 1. Dazu aber noch ein Hinweis: Derzeit wird 'Motor on/off' noch nicht von der Software des Monitors unterstützt, das heißt, der Motor läuft in beiden Varianten von BR3 ständig. Das ist unbedingt bei der Jumperwahl auf dem Floppy-Disk-Laufwerk zu berücksichtigen. 'Head load' darf nicht mit 'Motor on' verknüpft werden, ebenso darf kein ständiger 'Head load' auf der Controllerkarte gesteckt werden. BR2 ist also der Haltbarkeit von Schreib-/Lesekopf und Diskette zuliebe auf 2—3 zu jumpern.

Aus 'Gründen der Rationalisierung' wurde die Pfostenleiste BR6, die im Schaltplan vorhanden ist, auf der Platine eingespart, da sie elektrisch mit BR4 identisch ist. Leider sind auf dem Bestückungsplan keine Vorschläge für eine Standardverbindung abgebildet. Zunächst genügt eine Steckbrücke zwischen BR4/Pin 3 und BR5/Pin 3.

C7 (100nF) fehlt im Bestückungsplan in Heft 3, ist aber auf dem Bestückungsaufdruck der Platinen vorhanden. Er liegt zwischen RN3 und IC1, wobei ein Anschluß etwa auf der Höhe von Pin 1/RN3 liegt, der andere auf der Höhe von BR3.

C11 wird in der Stückliste mit 68µF angegeben, was einer Verzögerungszeit von 220ms (R6=4,7k) entspricht, einer durchaus realen Kopfladezeit. Ein 5,25"-Laufwerk einfacher Bauart kommt auf etwa 35ms. Dann wären rund 10µF erforderlich. Da R6 stets kleiner sein muß als 40k, wird man für C11 einen Elko nehmen wollen. Der negative Anschluß muß dann an R6 liegen, das heißt aber, auf der Platine weist der positive Anschluß zu R6.

Etwas verwirrend ist vielleicht die Funktion der vier DIL-Schalter. Da Mix-Betrieb (5,25/8") möglich ist, wird sich mancher fragen, wie S1 dann wohl zu setzen sei. S1 und seine drei Kollegen sind nur für den Boot-Vorgang relevant, die Schalterstellung ist anschliessend softwaremäßig überschreibbar.

Allgemeines

Inzwischen ist das Monitorlisting für Interessenten verfügbar. Die Mnemonics des verwendeten ASM 86 halten sich bei einigen Befehlen nicht an das Intel-Handbuch, der Lesbarkeit und dem Verständnis des Listings tut das aber keinen Abbruch.

Busplatine

Seit kurzem ist auch eine Busplatine mit zehn Steckplätzen verfügbar. Von einem Abdruck des Layouts haben wir aus den bekannten Gründen abgesehen, ohne professionelle Durchkontaktiertechnik ist hier nichts mehr zu machen. Das Vorhandensein von zehn Steckplätzen ist als Option zu verstehen. Zwei Versionen, eine mit fünf, eine mit zehn, hätten erhebliche Mehrkosten erbracht. Wer die zehn Steckplätze alle ausnutzen will, stößt bei 5 MHz Taktfrequenz allerdings bereits in kritische Regionen vor. Eine Busterminierung, der wir uns in absehbarer Zeit mit einem Beitrag annehmen werden, kann unter Umständen erforderlich werden.

Die vom Z80 her stammenden Leitungen des ECB-Bus IEI und IEO werden nicht verwendet, die Lage der Steckplätze sagt also nichts über Interrupt-Prioritäten aus.

Stromversorgung

Der eigentliche Computer benötigt nur ein Netzteil von 5 V/3 A (mehr Ampere können natürlich nicht schaden). Hilfsspannungen von +12 V und —12 V für Schnittstellen (RS 232) sind bei Leitungslängen in der Größenordnung von einem Meter nicht erforderlich, denn man kommt mit TTL-Pegel gut aus. Das hat natürlich nur dann Sinn, wenn die 'Gegenseite', also das Gerät, mit dem kommuniziert wird, ebenfalls mit TTL-Pegel zufrieden ist. In Verbindung mit dem c't-Terminal ist diese preiswerte Lösung leicht zu realisieren. Im reinen Terminalbetrieb benötigt man auch keine Handshake-Signale. Die Potentiale von $\overline{CTS}$ (IC7/Pin 17) und $\overline{DSR}$ (IC7/Pin 22) stellen sich 'von selbst' richtig ein.

Einbau

Der Einbau der Floppy-Laufwerke zusammen mit dem Rechner in ein Gehäuse ist, obwohl wir es gemacht haben, nicht unbedingt empfehlenswert. Die Beseitigung von Spikes auf der Versorgungsspannung bereitet dann einige zusätzliche Mühe. Wer das vermeiden will, sollte daher ein separates Gehäuse mit zwei eigenständigen Netzteilen für die Laufwerke einplanen.

Es hat sich außerdem gezeigt, daß eine gute Ventilation der Luft im Gehäuse unabdingbar ist. Bei Temperaturen oberhalb 70 Grad Celsius ist bereits mit 'Fehlleistungen' des Computers zu rechnen. Der Prozessor allein entwickelt bereits eine beachtliche Menge Wärme. Ein Lüfter ist bei Einbau in ein geschlossenes Gehäuse absolut notwendig.

Zum Abschluß noch etwas Erfreuliches: Die Ausführungszeiten von Programmen in MBASIC, das seiner Aufwendigkeit wegen sicher nicht zu den schnellsten gehört, haben uns schier verunsichert. Wenn nicht wirklich korrekte Ergebnisse angezeigt worden wären, hätten wir glatt bestritten, daß ein Rechner so schnell sein kann. □

Praxistip:

Der verflixte INPUT

beim CBM 8032

Michael Bauer

Was wäre ein Computer, wenn er fehlerlos arbeiten würde? Wahrscheinlich könnte so ein Rechner den 'Freaks' nur halb so viel Spaß machen. Als Spielverderber, die wir nun mal sind, zeigen wir mit diesem Praxistip, wie man den CBM 8032 um einen Fehler ärmer machen kann.

Vielen Anwendern des Commodore 8032 wird dieser Fehler im Betriebssystem des Rechners aufgefallen sein: Wartet der Computer in einer Input-Schleife auf die Eingabe eines Variablenwertes und drückt man nur die 'RETURN'-Taste, so springt der Rechner aus dem Programm.

Dieses unschöne Verhalten des CBM kann durch eine nur zwei Byte große Änderung im Betriebssystem behoben werden. Dazu muß man allerdings ein EPROM ändern und neu 'brennen'. Nach der Modifikation akzeptiert der Rechner die Eingabe von 'RETURN' und übernimmt den bestehenden Wert der Input-Variablen.

Vom Rat zur Tat

1. 4 KByte RAM im oberen Speicherbereich vor BASIC schützen:

```
10 POKE 53,112: CLR: A=28672
```

2. Den ROM-Inhalt in den geschützten Speicherbereich kopieren:

```
20 FOR I=49152 TO 53247:
     REM Für CBM 3032
21 FOR I=45056 TO 49151:
     REM Für CBM 8032
30 POKE A,PEEK(I): A=A+1
40 NEXT I
```

3. Den Code im RAM mit einem Monitor ändern:

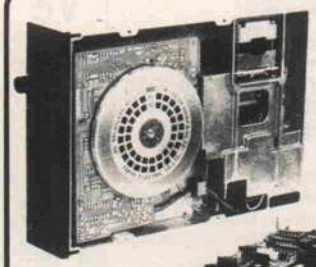
	Speicherzelle	Neuer Inhalt
CBM 3032	$ 7AF6	$ F0
	$ 7AF7	$ F2
CBM 8032	$ 7BF1	$ F0
	$ 7BF2	$ F2

4. Ein EPROM des Typs 2532 mit dem geänderten Speicherinhalt programmieren.

5. Computer ausschalten, Gehäuse öffnen, das ROM D5 (CBM 8032: UD10) entfernen und durch das EPROM ersetzen.

6. Funktion testen.

Wenn die Operation erfolgreich verlaufen ist, haben Sie sich als 'Freak' um eine Quelle der Freude gebracht und sich als 'normaler' Anwender ein Ärgernis ausgeräumt. □



CE-DATA®

Slimeline 5,25" FD Disk
250 KB - 1 MB
Einsetzbar für alle Computer Systeme

CE-DATA Slimeline Laufwerk 5,25"		
40 Track SS/DD, 250 KB		DM 575,-
80 Track DS/DD, 1 MB		DM 875,-
CE-DATA Doppelfloppy im Gehäuse, komplett betriebsfertig, a 250 KB		DM 1.398,-
Slimeline Floppy für Apple - halbschrittfähig		DM 625,-
Floppy Disk Controller für Apple (CE-DATA)		DM 115,-
Floppy Disk Controller für Apple (universal)		DM 189,-
Double Density Controller für Tandy und Video Genie		DM 198,-
Verbatim Disketten	ab	DM 49,-
BASF Disketten	ab	DM 48,-
Siemens PT 88 Tintenstrahldrucker		DM 1.750,-
Star Drucker Gemini 10 X		DM 975,-
Andere Star Drucker - Preis auf Anfrage		

Wir führen Floppy Disk Laufwerke, Interfaces für Tandy ☆ Video Genie ☆ Apple ☆ Triumph Adler

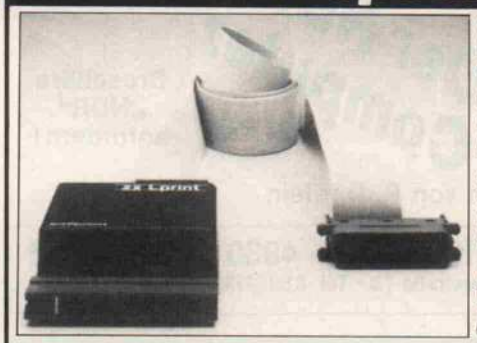
CE COMPUTER ELECTRONIC GMBH

Reichshofstr. 55
5840 Schwerte-Westhofen
Tel. 0 23 04/6 80 64–65

Kompletter Katalog gegen DM 5,- in Briefmarken
Händlerkonditionen bitte schriftlich erfragen!

CE-DATA Service löst Ihre Reparaturprobleme

NEU! RS 232 & Parallel-Druckerinterface für Sinclair Spectrum



Centronics und RS 232 in einem Interface

LPRINT, LLIST und COPY-Befehle

Ohne Zusatzsoftware funktionsfähig

Microdrive kompatibel

COPY in FARBE

Textverarbeitung Hochauflösende Grafik Listingdruck

Komplettpreis DM 198,-

Endlich können Sie mit Ihrem Sinclair Spectrum (16 und 48 K) auch professionelle Textverarbeitung realisieren.

Einfach das **ZX LPRINT III** *an die Rückseite Ihres Spectrum anstecken und alle Sinclair Druckbefehle inklusive COPY sind sofort auf Ihrem Drucker ausführbar.*

ZX LPRINT III *ist komplett. Sie benötigen keine umständliche Software mehr um mit Normalpapierdruckern arbeiten zu können.*

ZX LPRINT III *hat die notwendige Software im ROM. Das bedeutet für Sie: Anschalten, und Ihr Gerät ist betriebsbereit.*

ZX LPRINT III *stellt Ihnen wahlweise COPY-Routinen für alle gängigen Drucker zur Verfügung: z. B. Seikosha GP 100A und GP 250X, alle grafikfähigen EPSON Modelle, Star DP 510, DP 515 und STX 80, Star Gemini und Star Delta, Shinwa CP 80 und* **Seikosha GP 700 in voller Farbe.**

Die Copyroutinen erlauben die Benutzung selbstdefinierter Grafikzeichen und der Spectrum Blockgrafik. Bei Typenraddruckern und Schreibmaschinen werden bei Text-COPY nur die alphanumerischen Zeichen dargestellt.

Auch andere als die oben genannten Drucker oder Typenradmaschinen lassen sich mit **ZX LPRINT III** *benutzen, wenn sie über eine Standard Centronics Parallelschnittstelle oder RS 232 verfügen.*

ZX LPRINT III *ist voll kompatibel mit den demnächst erhältlichen Microdrives und zu Spectrum Programmen wie z. B.: Tasword, Viewfile u. a.*

Bitte geben Sie bei Ihrer Bestellung Ihren Druckertyp an. **ZX LPRINT III** *wird komplett mit Parallelanschlußkabel* **oder** *RS 232-Kabel und deutscher Anleitung geliefert. Sollten Sie beide Kabel benötigen erhöht sich der Endpreis um DM 50,-.*

Ihre Bestellung schicken Sie bitte mit Vorausscheck (plus 3,- DM für Porto) oder gegen Nachnahme (plus 8,- DM Nachnahmegebühr, Versand ins Ausland nur gegen Vorrausscheck) an:

MICROCOMPUTER LADEN

Computerpartner GmbH, Lietzenburger Str. 90, 1000 Berlin 15
Telefon 030/882 65 91, Telex 184 685 busa d

ZX LPRINT III gibts nur bei uns. Händleranfragen willkommen.

BMC Neue Produkte ab 1984:

BM 8181	Farbmonitor RGB 640 × 240 Pkt. für IBM, inkl. Kabel	DM 1584,60
BM 12 EN	hohe Bandbreite v. 20 MHz, grün, entspiegelt	DM 438,—
BM 12 EY	Bandbreite > 18 MHz, bernstein	DM 438,—
BM 12 A	Bandbreite > 15 MHz, grün	DM 298,—

BMC-Monitor, einzigartig in Qualität und Leistung, mit Filterscheibe, > 18 MHz, grüner Röhre, im formschönen Datenmonitor-Gehäuse

BM 12 ES = 398,— DM inkl. MwSt.
(349,12 DM netto) Dazu passend ERGOTILT 89,— DM inkl. MwSt.

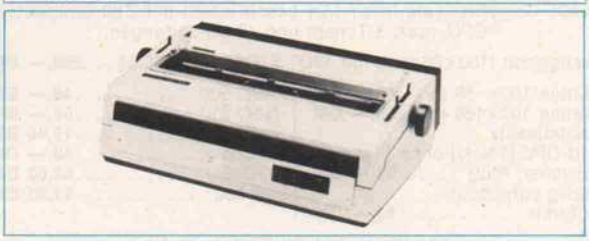
Neu !

HX 20 - Micro Terminal

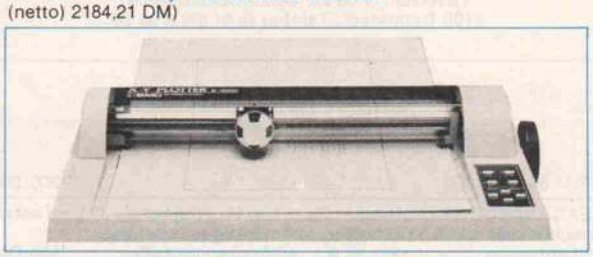
DM 1298,—

inkl. MwSt.

Dieses neue MICRO - TERMINAL für den EPSON HX20 Hand - Held - Computer gestattet die Darstellung von bis zu 80 Zeichen auf 25 Zeilen. Das 2000 Zeichen - Display mit grünem Schirmbild und Antireflexscheibe gewährleistet größtmögliche Benutzerfreundlichkeit. Sowohl Text, wie auch Graphik werden mit hoher Schärfe dargestellt. Eine hervorragende ergonomische Konstruktion gibt die Möglichkeit durch Drehen oder Kippen, das Sichtgerät auf optimalen Betrachtungswinkel einzustellen.



Low-Cost-Typenrad-Drucker
16 CpS, Schreibbreite 335 mm, 96-Z-Typenrad, Friktionsführung, mit Einzelblatteinzug, 8-bit-Schnittstelle 2-KB-Puffer
TD 16 = 2490,— DM inkl. MwSt.
(netto) 2184,21 DM)



Unser neuer unschlagbarer 4-Farben-Plotter
DIN A3-Format, 0,1 mm-Genauigkeit, Schreibgeschwindigkeit 100 mm/s, ASCII-Zeichensatz u. Kreisfunktion! Optional Graph-ROM!
MP 1003 2690,— DM inkl. MwSt.
(netto 2359,64 DM)

Händler-Rabatte auf alle Produkte ab dem 1. Stück!

mirwald electronic BMC

Fasanenstraße 8b, 8025 Unterhaching/München,
Telefon (0 89) 6 11 12 24, FS 5 213 476
Büro Frankfurt: Adalbertstr. 15
Telefon (06 11) 70 35 38

SOFTWARE + HARDWARE

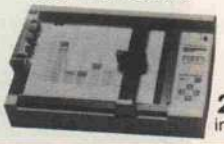
MANNESMANN-TALLY · HEWLETT PACKARD · Qume · IBM

MANNESMANN TALLY

Matrixdrucker MT-80
9*8 Matrix – Druckwegoptimierung – bidirektionaler Druck – 80 Zeichen/sek. – Grafikfähig – Endlos- und Einzelblattverarbeitung – Centronics 8 bit parallel – seriell V. 24 optional

DM **1098,–** incl. MwSt.

Pixy-Plotter
3 Farben Plotter integrierter Microprozessor 200 mm/sek. Centronics 8 bit parallel – seriell V. 24 optional



DM **2650,–** incl. MwSt.

WITTKE + RIEGRAF

Gesellschaft für System- u. Software-Entwicklung mbH
Postfach 3107 · Joh.-Strauß-Straße 7
7024 Filderstadt 3
Telefon 0 71 58/6 40 14 · Telex 721 296

VC P | Völzke Computer Peripherie | VC P

Info EU gegen Rückporto

Eprom-Programmer V 128 für VC-20, C-64 und EX-64 für Eproms 2508/16/32 und 2758/16/32/64/128. Professionelle Ausführung mit komfortabler Treiber-Software auf Kassette: ..... **DM 265,–**

UNIMENT-C 64-Befehlserweiterung: über 50 zus. Befehle und Funktionen für Assembler, Centronics-Interface, Graphik-, Sprite-, Sound- u. Disketten-Anwendung; mit Beispielprogrammen u. ausführl. Bedienungs-Anleitung.
Kassette ..... **DM 99,–**
Diskette zuzgl. ..... DM 8,–
UNIMENT-Steckmodul. ..... **DM 199,–**

Weiteres aus unserem Programm:
- Epromkarten für VC-20 und C-64
- Speicherkarte für VC-20 mit 32 KByte RAM und Steckplatz für 16 KByte Eprom

Hagen Völzke, Ahornallee 4, 8023 Pullach
Telefon 0 89 / 7 93 45 34 Händleranfragen erwünscht!

RATEV ELEKTRONIK-VERTRIEBS GMBH
Postfach 16 01, 4030 Ratingen 1, Tel. 0 21 02 / 2 99 02

Hochwertige Cherry-Tastatur, wie Titelbild c't Ausgabe 1, mit 8 bit parallelem Datenausgang, die an jedes System oder jeden Selbstbaucomputer angepaßt werden kann, da die ursprüngliche Version mit einem EPROM für alle Anwendungen veränderbar ist.
Eine ausführliche Umbauanleitung liegt vor.
Preis inkl. Gehäuse ..... **DM 390,—**
dto., jedoch für APPLE bereits umgerüstet ..... **DM 449,50**

Für TRS 80 ● Video-Genie ● Apple

Eprom-Programmiergerät SE 40
für 2716/2732/2532/2758, kompl. anschlußfertig, Software auf Kass. oder Disk **Preis 269,— DM**

Eprom-Löschgerät SE 50
für max. 5 Eproms, Löschdauer ca. 10 Min.
Preis 125,— DM

afu electronic vertriebs gmbh

Steinstraße 9, 5778 Meschede, Telefon 02 91/75 85

CEPAC-80

CMOS-Computer (wie in c't 1/84 beschrieben) mit Z80-kompatibler CPU, max. 3 Timern und 48 I/O-Leitungen.

Fertiggerät (100 x 86 mm) mit RIOT 81C55 und 27C64 ... **258,— DM**

Platine 100 x 86 mm	**49,— DM**	NSC 800	**48,— DM**
Platine 100 x 160 mm	**69,— DM**	NSC 810	**68,— DM**
Bauteilesatz (Mit CPU [1 MHz] ohne Platine, Speicher, Riot)	**99,— DM**	8155	**19,80 DM**
		81C55	**48,— DM**
Fertig aufgebaut, Aufpreis	**40,— DM**	27C16	**42,50 DM**
		NE 590	**11,80 DM**

Alle Preise inkl. MwSt. Info gratis.

CONITEC **Christian Lotter KG**

Postfach 11 06 22, Schuchardstraße 4
6100 Darmstadt, Telefon (0 61 51) 2 60 13

Mikrocomputerkurs im Schulfernsehen (Nord III)

Mit Z 80 und 68008, Vollgrafik und Roboter

Der NDR-Klein Computer

Broschüre „NDR" anfordern!

Mit Begleitbuch von R. D. Klein

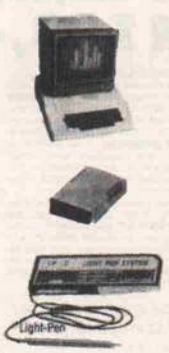
ELEKTRONIKLADEN · 4930 Detmold 18
Eggestraße 70 · Tel. 0 52 32/81 71

DR. AUMANN

CP/M 86 f. PC/XT			205 DM
TEAC-Floppy-Disk-Laufwerke: Ganz neu im Programm, 3"-Laufwerke anschließbar wie 5"-Laufwerke, vollständig kompatibel.			
3"-Laufwerke	FD 30 A	40 Spuren einseitig	550 DM
5"-Laufwerke	FD 55 A		640 DM
(alle in SLIMLINE-Ausführung)	FD 55 B		750 DM
	FD 55 E		740 DM
	FD 55 F		880 DM
	FD 55 G	1,66 Mio. Byte	1040 DM
Genie 16, IBM-kompatibel 8086, 256 K, 2 x 360 KB Disk, HyRes-Graphik, flache freibewegliche Tastatur			5900 DM

Markendisketten ab DM 5,47 (10er-Preis) mit Verstärkungsring. Wir liefern auch Steckverbinder, Bauteile, Drucker zu interessanten Preisen. Alle Preise incl. MwSt., Versand ab Straßenhaus.
Fordern Sie unsere Infos an!!!

Dr. Aumann GmbH Computersysteme, Schulstraße 12, 5451 Straßenhaus
Tel. 0 26 34/40 81, Bürozeiten: montags — freitags 9—12/14—17 Uhr



Händler gesucht

Und hier bekommen Sie:

Proficomputer mit zwei CPU's und **64 K RAM + 12 K ROM** **CPU 6502 + Z80 A** auf dem Motherboard, Int.-Basic + Monitor prog., 7 Slots, alle IC's gesockelt, betriebsbereit
Farbe / Grafik apple-compatibel + CP/M-fähig **DM 1265,–**
Light-Pen: High-Resolution, Interface + Software nur **DM 519,—**
Graphic-Table: Software + Hardware, Plott-II kompl. **DM 269,—**
Slimline Disc-Laufwerk 5 1/4 Zoll, FDD-820 ein Qualitätslaufwerk von TEAC. Im Gehäuse, voll apple-compatibel .... **DM 685,—**
Disc-Controller-Card ..... **DM 135,—**
Disketten höchster Qualität, 1S/125K-40Tr. 10 Stk. nur **DM 49,50**
RGB-Monitor 12", Color, 18 MHz, unbegrenzte Farben **DM 900,—**
Color-Monitor 14", NTSC-Eingang, für Apple-Computer ..... nur **DM 690,—**
Monitor 9" grün, High-Resolution 20 MHz, bei uns nur **DM 279,—**
IBM-compatible in Kürze lieferbar.

Weiteres COMPUTER-Zubehör finden Sie in unserer SONDERLISTE, kostenlos anfordern!!!
Preise incl. MwSt. + Versand. Apple ist eingetr. Warenz. der Apple Inc.

HÖSCH Elektronic Bruchstr. 43 4000 Düsseldorf 1 Tel. 0211 / 67 62 14

NEU

Die perfekte Erweiterung Ihres CBM64 ... mit Zukunftperspektiven

- Digitale Uhr
- Mischmöglichkeit { Grafik-Text, auch in Farbe / Haarscharf und stabil
- Grafische Darstellung; in den Hintergrund umschaltbar
- Oberste Reihe kann festgesetzt werden.
- Einstellbarer Zeilenabstand.
- Nimmt keine Speicherbereich in Anspruch!
- Ton bleibt voll erhalten.
- Keine extra Stromversorgung erforderlich.
- Videospeicher des **CBM64** kann frei zugeschaltet werden.

GRATIS DAZU! AUSSERGEWÖHNLICHER DEUTSCHER TEXTVERARBEITER mit umfangreicher deutscher Betriebsanleitung

DIE 80-ZEICHEN / GRAFIKKARTE DM 279,-

★Warnung: Passen Sie auf vor (mangelhafte) Kopien von ROOS-Produkten!

für VC 20 und CBM 64

40/80 ZEICHENKARTE
40/80 Zeichen p/z, haarscharf und stabil. **DM 249.-**

64kRAM + 2k EPROM
Mit schneller Software für RAM-Files **DM 279.-**

EPROMKARTE **DM 45.-**
Zwei 4k EPROM Sockel. Adressen einstellbar

STECKPLATZ-ERWEITERUNGEN
2 Steckplätze **DM 69.-**
5 Steckplätze, Vollpufferung, 5V Stromversorgung, Ein-Ausschaltbar. VC20 **DM 139.-** CBM64 **DM 169.-**

EPROM PROGRAMMIERER
für 2716, 2732, 2764, 27128, 2532. anzuschliessen an jeden Microcomputer. Mit Software, **DM 174.50**

EPROM-LÖSCHGERÄT
löscht 4 EPROMS gleichzeitig. **DM 120.-**

EPROM-BANK 128k Eprom
für TRS 80 en VIDEOGENIE **DM 379.-**

MACH3 DRUCKER-PUFFER

	16 k	32 k	48 k
par. ein — par. aus	389.-	445.-	499.-
serie in — par. aus	431.-	497.-	559.-

Für all unsere Produkte: **Händler Anfragen erwünscht.**

ROOS ELEKTRONIK

- Alle Preise einschliesslich MwSt.
- Versand per Nachnahme oder Vorkasse.
- Von al unseren Produkten haben wir ausführliche Prospekte, die wir Ihnen gerne kostenlos zusenden.
- Alle Geräte können freibleibend ausprobiert werden. Unbeschädigt innerhalb von 10 Tagen zurückgesandt, bezahlen Sie nur Verpackungs- Versandkosten.

KLEINER MARKT 7 U + 4190 KLEVE + TELEFON 0 28 21 / 2 88 26

computertechnik G. WEBER

Eulenspiegelstr. 56, 8000 München 83, 089/601 2554

NIEDRIGSTPREISE

SHARP	HEWLETT-PACKARD	EPSON
-alle PC's	-HP 10er Serie	-HX-20
-MZ-700	-HP 40er Serie	Microterminal
Pascal, Ass.	-HP 75er Serie	für HX-20
Floppys	APPLE	-Drucker
-35xx Serie	Super Paketangebote	-QX-10

Zubehör für alle Produkte
Schnittstellen, Kabel, Floppys, Markendisketten
Drucker mit vollem Zeichensatz für SHARP MZ-700

SIDNEY II

hib

HIB GMBH
MICROCOMPUTER
Wunderburggasse 2
8500 Nürnberg 1
0911 / 22 13 83

SIDNEY - die Computer für den professionellen Anwender!

ausgestattet mit 2 Prozessoren
SIDNEY II: 6502 und Z-80
SIDNEY 88: 6502 und 8088
64 K RAM aufrüstbar bis zu 512 K
incl. PROFI-Tastatur ab DM 3640,00

ALLES für den APPLE II

PROFI-Tastatur, einzeln	DM	698,00
128 K RAM Karte (Saturn compatibel)	DM	498,00
5" Flachlaufwerk anschlußfertig	DM	589,00
Plot-II Graphics Table	DM	269,00
SUPER JOYSTICK	DM	118,00
Minidisks, "CONTROL DATA", SS/DD 10 St. ab	DM	49,00
Diskettenbox für 80 Mini-Disks, mit Schloß	DM	59,00

Alle Preise incl. Mehrwertsteuer
Gratispreisliste anfordern!

Wiederverkäufer-Preise nur auf schriftliche Anfrage.

FUTTER für den C64

Ihr Computer ist ohne Programme wie ein Auto ohne Benzin.

Gute und preiswerte Programme für Ihren C64 bieten wir mit dem SYNTAX-Programm-Kassetten-Magazin.

Jeden Monat erscheint eine Kassette mit 6 neuen, vielseitigen Programmen für Ihren C64.

SYNTAX-Programme auf Kassetten und Disketten sind auch für die Commodore CBM und VC 20 erhältlich. Nutzen Sie Ihr Gerät verstärkt durch neue Ideen.

Fordern Sie gleich heute noch unter Angabe Ihres Gerätetyps kostenlose Informationen von

SYNTAX

Soft- u. Hardware GmbH
P.B. 18 68, 7550 Rastatt
Telefon (0 72 22) 7 23 15

ZX 81 und ZX Spectrum
Zubehör von Logitek

Druckerinterface für ZX-Spectrum
Anschlußfertig für fast alle erhältliche Drucker wie EPSON, STAR, CP80 usw. LPRINT, LLIST und COPY sind sofort verfügbar. Viele zusätzliche Druckfunktionen sind in BASIC ansprechbar, wie wählbare Schriftstärke und Schriftbreite, direkte ASCII-Ausgabe über LPRINT, Erweiterung des Spectrumzeichensatzes mit deutschen Zeichen, in Verbindung mit der 80 K Speichererweiterung ist während des Druckens zugleich ein Weiterarbeiten am Rechner möglich.
Komplett mit Kabel **DM 198,—**

LOGITEK Spectrumgehäuse
Dieses formschöne schwarz eloxierte Aluminiumgehäuse nimmt den Spectrum mit Netzteil und Busplatine für 5 Erweiterungen auf. Mit EIN/AUS Schalter.
Gehäuse komplett mit Bus **DM 169,—**
Busplatine für 5 Karten **DM 89,—**

Speichererweiterung von 16 K **auf 80 K** zum Einstecken in den Spectrum. **DM 198,—**
(Bitte bei Bestellung Issue 2 oder 3 angeben!)

32 Bit Portmodul
Über die 32 Leitungen lassen sich elektronische Steuer-, Regel- und Meßschaltungen anschließen, die digitale Ein- und Ausgänge haben.
Für ZX-Spectrum und ZX-81 **DM 138,—**

Spectrumstecker **DM 14,—**
Gegenstück dazu **DM 7,—**

64 K RAM Modul für ZX-81
schwarz elox. Alugehäuse, flach an den ZX-81 ansteckbar, Port durchgeführt. **DM 210,—**

ZX-81 Stecker **DM 12,—**
Gegenstück dazu **DM 6,—**

Räumliche Grafik für ZX Spectrum
Mit diesem Programm lassen sich mathematische Funktionen räumlich darstellen und anschaulich erklären. **DM 38,—**

Weitere Programmkassetten auf Anfrage.

Deutsche Beschreibungen werden mitgeliefert. Preise incl. MwSt. Versand per NN zzgl. 6,50 DM. Porto und Verpackung ab Lager Berlin.

LOGITEK
Andreas Höft und Frank Lesser GbR
Pankstraße 49, 1000 Berlin 65
Telefon (030) 4 61 64 92

Hardware für cbm

Hochauflösende Grafik mit 521424 Bildpunkten auf 4 unabhängig. Bildschirmseiten inkl. 8 KByte komfortabler Grafiksoftware im EPROM. Comgraph 8000 professional ab **898.32 DM**

Grafik 3D — professionelles Programm zum Erstellen von dreidimensionalen Grafiken **498.00 DM**

Linien-, Balken- und Kreisdiagramme auf allen Commodore 8032/96 durch Programmpakete inkl. intelligenter Softw. mit Dialogsteuerung **198.00 DM**

ROM-Karten erweitern die EPROM-Steckplätze im cbm um ein vielfaches ab **56.00 DM**

Softrom 4 KByte — die komfortable Programmierhilfe. Natürlich lötfrei einsetzbar für **188.10 DM**
(umsch.bar als 2716/2732/2532-Ersatz)

Epromer f. jeden cbm m. leist.f. Menü-Softw. Qualitätsprodukt: **329.46 DM**

64 KByte RAM-Erweiterung, voll Los-96-kompatibel **738.72 DM**

Interface cbm-Centronics — endlich eines, das richtig funktioniert. Einsetzbar für alle Itoh-, Epson- und andere Drucker. Bietet die Möglichkeit 2 Datenkanäle gleichzeitig offenzuhalten. Industrie-Qualität. **259.92 DM**

Hardware für IBM-PC

Z.B. Prototype-Board zum Aufbau von eigenen Schaltungen mit vergoldeten Kontaktzungen durchkontaktiert. Ein gut durchdachtes Board in bester Ausführung ab **98.00 DM**

EEPROM-Karte, RAM-Karten etc.

Drucker

Schnelle Matrixdrucker: Itoh 8510A **1698.00 DM**

Siemens-Tintendrucker mit Centronics- oder cbm-Schnittstelle, komplett **2298.00 DM**

Farbb. f. Epson MX-80/FX-80: **17.50 DM**
Farbb. für Itoh 8510/1550: **16.90 DM**

Disketten d + b Disky 3204 SS/DD 10 St. **69.00 DM**

dto. für cbm 8250 77 Spuren, gepr. 10 St. **139.00 DM**

Alle Preise inkl. MwSt. Ausführliche Informationen vom Hersteller:

Händleranfragen erwünscht.

Rösch Datentechnik GmbH
Software- und Systementwicklung
—Industrieelektronik—
Jasperallee 41, D-3300 Braunschweig
Tel. 0531/336046, Tlx 952759 rodat-d

NEU CBM

ALLE PREISE INCL. 14 % MWST.

CBM 80XX

HIGH-SPEED-GRAFIK: mit Soft-Pen und 4 Bildschirmseiten für 8032, lötfreier Einbau, 8K Betr.-Software in 2 Proms 4 x 128.000 Punkte Auflösung ..... **1298,—**
GRAFIK-SOFTWARE/BADI: Balkendiagramme für 1 + 2 Jahre und um 12 + 24 x freie Balken zu definieren. Paketpreis .. **565,—**
ROM-ERWEIT.-KARTEN: Erweiterung der freien Plätze auf 2 x 7 Steckplätze **348,—**
auf 1 x 3 Steckplätze ..... **161,—**
auf 1 x 2 Steckplätze ..... **128,—**
ITOH-DRUCKER 8510 A ..... **1921,—**
ITOH-CBM-INTERFACE ..... **399,—**
EXBASIC LEVEL II: Superutility für CBM in 2 Ep-Roms, mit Handbuch ..... **380,—**
ROM-LISTING: Für CBM-700-Serie **99,—**
80/40-EPROM: 3000er Basic-Programm (mit poke + peek), laufen auf jedem CBM-8000 ..... **211,—**
PETSPEED-BASIC-COMPILER: 1. Optimierender Basic-Compiler/CBM-8000 **1100,—**
DATEX-2000: ein Superprogramm für die anspruchsv. Textverarbeitung .... **759,—**
WCS-ADRESS: Adressverwaltung + Datenbank Select. + Sort. nach ALLEN Krit. Zugriff auf ca. 1800 Adr. in 10 Sek. **399,—**
LISTSCHUTZ-AUTORUN: Perfekter Schutz Ihrer Programme und Ihres (C) .. **359,—**
CBM-LEITFADEN: für CBM-8000/3000 (Basic 4.0) 1000 Tips + Tricks .. **248,—**

C64

SERIELL-PARALLEL: Drucker-Interf. **316,—**
IEEE-488-BUS: UNEINGESCHRÄNKT ALLE C64-Programme laufen m. d. 8250/8050/4040 ..... **316,—**
GRAFIK-TABLETT: Mit der VOLLEN Auflösung d. C64. Fertige Grafiken können in eigene Programme übernommen werden! ..... **348,—**
Und ... und ... und ...

Porto + Verpackung 8,70/lm Versand nur Nachnahme oder Vorkasse. **ALLE** Preise incl. der ges. MwSt.

Fordern Sie bitte unsere **INFO** an:

WCS-WEMPER COMPSOFT
BÖRNSENER STR. 6, 2055 AUMÜHLE
TEL.: (0 41 04) 53 65

Wordcraft 40 for the CBM 64

Buglass & Long
Heinrichstraße 20
3000 Hannover 1
Steckmodul
für C-64 und SX-64:
Preis 345,00 DM

Textsysteme für kleine Computer gibt es mittlerweile wie Sand am Meer — wenn da noch jemand mit einem neuen Programm kommt, muß es entweder überflüssig oder schon etwas Besonderes sein.

Der Vorteil des Programmes liegt zunächst in seiner Kompatibilität: Es gibt Versionen für alle Commodore-Rechner, den VC-20 eingeschlossen, und damit lassen sich auf dem C64 erstellte Manuskripte auch auf anderen Rechnern, zum Beispiel einem 8032, weiterbearbeiten. Das im übrigen in Form eines 50seitigen Lehrbuches verfaßte, sehr übersichtliche Handbuch zum Wordcraft 40 beinhaltet ein entsprechendes Konvertierungsprogramm. Leider hat der C64 nur einen 40-Zeichen-Bildschirm, aber das stört Wordcraft nicht: Man kann auf 40 Zeichen schreiben, seinen Text zu jedem gewünschten Zeitpunkt auf das endgültig gewünschte Format (von 5 bis 117 Zeichen Breite) umstellen und auch so ausdrucken lassen. Wählt man die Zeilenlänge größer als 40 Zeichen, dann bewegt sich der Bildschirm als 'Fenster' über den Text. Seitenformate und Textlängen lassen sich definieren, automatischer Randausgleich und Zentrierung, beliebige Tabulatoren auf Textanfänge oder Dezimalstellen (für Zahlenkolonnen), Einrücken, Ändern, Ergänzen, ist möglich. Angeschnittene Wörter gibt es nicht, so werden selbsttätig in die neue Zeile herübergezogen. Mischen oder Zusammenfügen verschiedener Texte, Fettdruck oder automatische Unterstreichung: Man findet selbst nach monatelangem Gebrauch noch immer neue Möglichkeiten. Adreßdateien lassen sich integrieren, der 'persönliche Serienbrief' ist kein Problem. 'Leider' ist hier die Software komfortabler als die Hardware, denn viele der (Ausgabe)möglichkeiten lassen sich mit dem Standard-Commodore-Equipment gar nicht nutzen, weil zum Beispiel der Drucker 1526 diese gar nicht kennt.

Wordcraft läuft sowohl mit Cassette als auch mit (einer oder mehreren) Diskettenstationen und Druckern über den Commodore-Bus, Parallel- oder RS-232-Schnittstelle. Dabei werden Parität und Baudrate von der Software abgefragt.

Fazit: Ein gutes und solides Handwerkszeug; im Zeitalter der 99-Mark-Programme erscheint es etwas teuer. Durch die gebotene Aufwärts- und Abwärtskompatibilität (es gibt sogar Versionen für CP/M, den IBM PC und Sirius) ist es jedoch seinen Preis wert, zumal zu erwarten ist, daß es auch für die zukünftigen CBM (264, ...) eine adaptierte Wordcraft-Version geben wird.

ES

C.C.Software, USA

M/PC : CP/M 2.2 Source Generator

TESCO GmbH
Rüdenhausener Straße
8714 Wiesentheid
Tel. 09383/1237
Diskette 8"
Preis: 199,00 DM

M/PC ist ein Quellcode Generator, der von der Firma C.C.Software, USA entwickelt wurde. Dieses Programm wird im deutschsprachigen Raum von der Firma TESCO GmbH vertrieben. M/PC erzeugt aus dem CP/M 2.2, das im Arbeitsspeicher des Rechners steht, ein ausführlich kommentiertes 8080-Source-Listing von circa 100 KByte Länge. Alle Datenbereiche innerhalb von CP/M 2.2 werden durch M/PC vollständig erweitert (z. B. 'BDOS ERROR ON A:'), und alle Labels innerhalb des Betriebssystems erhalten sinnvolle Namen (z. B. 'BITMAP').

Das von M/PC in etwa zehn Minuten erzeugte Source-Listing kann anschließend editiert, geändert und mittels eines geeigneten 8080 Assemblers wieder assembliert werden.

Das uns zur Verfügung gestellte Programm wurde auf einer 8"-Diskette (SS,SD) geliefert. Um die Arbeit von M/PC zu testen, wurden drei verschiedene Betriebssysteme CP/M 2.2 (64 KByte) von drei Besitzern als Grundlage für ein Source-Listing verwendet. Erstaunlicherweise trat dabei immer an der Adresse F6D2h (BASE: E400h, BIOS:FA00h) ein Übersetzungsfehler auf.

Um herauszufinden, ob M/PC auch modifizierte Versionen des CP/M 2.2 richtig übersetzt, wurde das Unterprogramm 'UPPER' (wandelt Kleinbuchstaben in Großbuchstaben) leicht verändert und ein neues SOURCE-Listing erzeugt. Es zeigte sich, daß die Modifikationen von M/PC nicht berücksichtigt wurden. Sollte etwa M/PC keinen echten Disassembler, sondern nur ein elegant getarntes Listing darstellen? Es sind offenbar zumindest einige Teile des CP/M Source-Codes in den Datenfiles von M/PC versteckt.

Wenn das CP/M Source-Listing wieder in Maschinencode verwandelt werden soll, können sich Probleme ergeben. Soll die Übersetzung von anderen als den angegebenen Assemblern gemacht werden, müssen die von M/PC vergebenen Labelnamen mühsam geändert werden. Die verteilten Namen sind teilweise bis zu acht Zeichen lang, unterscheiden sich aber in den ersten sechs Stellen. Das stellt für die in der Dokumentation angegebenen Assembler kein Problem dar. Es gibt aber noch andere Assembler, die achtstellige Labelnamen nicht verarbeiten können.

Auch fehlt hinter den Labels ein Doppelpunkt. Die vorgeschlagenen Assembler werden hierdurch nicht gestört, andere aber benötigen den Doppelpunkt (z. B. M80.COM).

Anwender, denen eine Diskettenkapazität von weniger als 241 KByte zur Verfügung steht, können M/PC nicht betreiben. Das Programm belegt allein 155 KByte auf der Diskette. Dazu kommt noch das erzeugte Source-Listing mit einem Platzbedarf von circa 100 KByte.

Obwohl M/PC einige Schönheitsfehler aufweist, kann es jedem Anwender empfohlen werden, der sein CP/M verändert möchte und dazu ein sehr gut kommentiertes Source-Listing benötigt. Vorausgesetzt, daß der Anwender den recht hohen Preis von 199 D-Mark für den einmaligen Gebrauch nicht scheut.

Nach den Änderungen am CP/M sollte man sich aber nicht mehr auf ein mit M/PC generiertes Listing verlassen.

GU

Udo Bartz

MYSTERY

Verlag Heinz Heise GmbH
Bissendorfer Straße 8
3000 Hannover 61
Kassette für ZX81
Preis 19,80 DM

Spaß und Spannung im MYSTERY-Land verspricht ein neues Spiel für den ZX81 mit mindestens 16-KByte-Speichererweiterung. Es handelt sich nicht um eines der infantilen Knall- und Schießspiele im Weltraum-Millieu, sondern um ein interessantes Abenteuer-Spiel, das vom Spieler logisches Denkvermögen und Gedächtnis-Leistungen verlangt.

Zum Spielablauf: Die Aufgabe besteht darin, mit einer Spielfigur den Ausgang eines aus Wegen und Brücken bestehenden Labyrinths unbeschadet zu erreichen. Als 'Schlüssel' zum Passieren des Ausgangs benötigt der Spieler eine Anzahl von Gegenständen, die er im Verlauf des Spiels erwerben muß. Dazu sind viele Abenteuer zu bestehen: Skurile Spukgestalten stellen sich dem Wanderer in den Weg. Die gefundenen Gegenstände, aus denen der Spieler eine wohlüberlegte Auswahl treffen muß, wechseln plötzlich die Gestalt.

Der recht spannend gestaltete Spielablauf verspricht einen hohen Unterhaltungswert. Der Schwierigkeitsgrad ist so gewählt, daß der Spieler etliche Anläufe unternehmen muß, um das Ziel zu erreichen. Eine Garantie für ein zweites Gelingen besitzt er dann noch lange nicht, denn bei jedem neuen Spiel ändern sich die Bedingungen.

Die Kassette besteht aus einer 'INFO' — einer ausführlichen Erläuterung der Spielregeln in deutscher Sprache — und dem Hauptprogramm. Dieses ist eine gut gelungene Mischung aus Text und Grafik und nutzt die eingeschränkten Darstellungsmöglichkeit des kleinen ZX81-Computers sehr gut aus.

EB

Preisgünstig zu CP/M86

Ein fertig angepaßtes CP/M86 für den c't86 ist nicht billig. Das liegt daran, daß die relativ hohen Lizenzkosten bei geringen Stückzahlen voll zu Buche schlagen. Es gibt aber einen Weg, preisgünstig zu einem angepaßten Betriebssystem zu kommen: c't-86-Anwender können eine CP/M-86-Version für den IBM PC kaufen, die im Handel schon für rund 200 Mark erhältlich ist. Beim Software-Service des Heise-Verlags kann man das BIOS für den c't86 sowie das Boot-Programm, das Formatierungsprogramm und das Programm Systemcopy auf die Original-Diskette kopieren lassen. Ergebnis: Ein CP/M-86-Betriebssystem für den c't86. Der Kopierdienst kostet 20 Mark.

Wer diesen Service nutzen will, sollte die System-Diskette(n) in ausreichend fester Verpackung an den Heise-Software-Service, Bissendorfer Straße 8, 3000 Hannover 61, senden. Legen Sie bitte einen Verrechnungsscheck über den Betrag von 20 DM und einen Zettel mit genauer Angabe des Diskettenformats bei. Erlaubt sind folgende Formate:

5¼-Zoll, 40 Track, SS oder DS, DD, 8 oder 9 Sektoren pro Track und

8-Zoll, SS, SD (IBM-Standard-Format 3740).

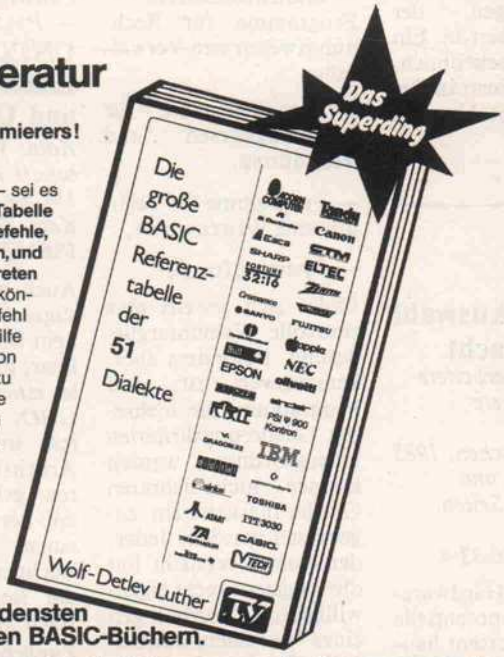
Das Einkopieren der genannten Programme erfolgt ohne Gewähr für etwaige Beschädigungen.

COMPUTER-Software-Literatur

Die Referenztabelle eines jeden BASIC-Programmierers! Unentbehrlich für Konvertierungen!

Wo immer Sie das BASIC-Listing eines Computers finden – sei es in Zeitschriften, Büchern, Clubmagazinen etc. – **mit dieser Tabelle können Sie alle rechnerspezifischen Sonder- und Grafikbefehle, Ein- und Ausgabebefehle für Bildschirm, Drucker, Kassetten, und Disketten, Funktionen und Systembefehle in ihrer konkreten Anwendung nachschlagen.** Bei Konvertierungsarbeiten können Sie sofort den für Ihren Computer zutreffenden Befehl ablesen. Computerumsteiger und Neulinge können mit Hilfe dieser Tabelle den Rechner ausfindig machen, der den von Ihnen benötigten BASIC-Befehlsvorrat hat, so daß die zu lösenden Probleme auch bewältigt werden können. Die große BASIC-Referenztabelle ist auch die große Hilfe im BASIC-Unterricht, da sie eine **bisher nicht dagewesene Vollständigkeit von BASIC-Dialekten im Zusammenhang** bietet.

1375x980mm **patentgefaltet** (1,3475 m²) und 96 Seiten Format 144x278mm – Bestell-Nr. LV-033-X · 45,– DM



Umfangreiche Software (Listings) mit ausführlicher Dokumentation der verschiedensten Anwendungsbereiche finden Sie in unseren BASIC-Büchern.

Praxisgerechte Computer-Literatur für Geschäft und Privat

Im Fachhandel Prospekt TE2 gegen Freiumschlag

W.-D. Luther-Verlag

Elisabethenstraße 32 · 6555 SPRENDLINGEN

Drucker

gibt es viele Typenrad- u. Nadeldrucker von: **Epson, Olivetti, Nec, Qume, Seikosha, Star, Brother, Juki, Mannesmann, Philips** ...

Doch

den Richtigen für SIE und Ihr Problem gibt es bei uns. Über Superpreise reden wir nicht. Wir haben Sie!

Computer

große und kleine, bekannte, weniger bekannte, ob **Televideo, Epson, Genie, Kaypro, Olivetti,** ob unser ASS **'Axio'**, ... Fragen **SIE** nach **Ihrem** Computer, er wartet darauf, Ihnen sein Können zu demonstrieren. Geballte Leistung zu unglaublichen Preisen.

Software

darüber reden viele, wir haben **die,** die Ihre Probleme löst.

Zubehör

Disketten-Kassetten-Bänder, Papier-Boxen-Farbbänder, **traumhafte Preise**

Wirklich nicht zu glauben, darum:

Anwender — Freaks — Händler

für jeden das RICHTIGE zu ..... Preisen??

Drucker Computer Zubehör

S.T.A. Data Control Corp.
Türmergasse 25 · 6900 Heidelberg
Telefon 0 62 21 · 78 05 55

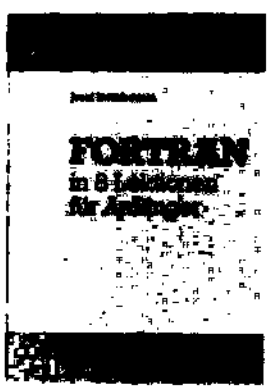
STA

Josef Kwiatkowski

Fortran

in 8 Lektionen für Anfänger
Stuttgart, 1983
Frech-Verlag
216 Seiten,
kart. DM 29,80
ISBN 3-7724-5428-3

Während die Computer-Hardware eine Revolution nach der anderen durchmacht, einerseits immer schnellere, immer größere Maschinen gebaut werden, andererseits die Personal-Computer immer leistungsfähiger und billiger werden, zeichnen sich im Softwarebereich Fixpunkte und Traditionen ab: Die Programmiersprachen als Schnittstelle zum menschlichen Bediener. So die alte Sprache Fortran aus dem Jahre 1954, die sich nicht nur 30 Jahre in älteren und modernen Versionen auf Großrechenanlagen in technisch-naturwissenschaftlichen Aufgaben bewährt hat, sondern auch seit Jahren schon im Mikrocomputerbereich viele Anwender gefunden hat. Fortran wird in diesem Buch dem Anfänger vorgestellt, von dem nicht das Verständnis komplizierter mathematischer Sachverhalte verlangt wird.



In 8 Lektionen mit 77 Übungsaufgaben wird er mit den Grundelementen dieser Sprache vertraut gemacht; mit Konstanten und Variablen, Ein- und Ausgabeanweisungen, Steuer- und Schleifenanweisungen. Nun ist Fortran eine Sprache, die es dem Benutzer nicht gerade leicht macht. Dem Autor gelingt es aber, in gut durchdachten, mit übersichtlichen Tabellen, Listings und Flußdiagrammen ausgestatteten Kapiteln, den Anfänger schließlich dazu zu führen, selbst Funktionen zu programmieren, mit denen er zum Beispiel die Fakultät einer Zahl berechnen kann. Es schließen sich ein kleines Lexikon der EDV-Fachausdrücke und letztlich die Lösungen der Übungsaufgaben an. Ein kleines Taschenbuch, daß jedem Fortran-Beginner zu empfehlen ist.

AB

Peter Knorr

Software-Auswahl leicht gemacht

2. völlig überarbeitete und aktualisierte Ausgabe.
Haar bei München, 1983
Verlag Markt und Technik. 423 Seiten,
DM 58,—
ISBN 3-922120-33-4

Nach einem Hardwarekauf wird der potentielle Softwareinteressent häufig vom Hardwareverkäufer alleingelassen. Er sieht sich zwar einem umfänglichen Programmangebot gegenüber, gilt es jedoch, seine speziellen Wünsche zu befriedigen, verwischt sich der Eindruck der enormen Softwarefülle schnell.

An diesen Interessentenkreis wendet sich das Buch, das circa 2000 Programme für Personalcomputer vorstellt. Zu jedem Programm werden 'Programmbeschreibung', 'Hardwarebeschränkung', 'Preis', 'Zahlungsweise', 'Autor' und 'Bezugsquellen' angegeben. Diese durchgängig verwandte Gliederung in der Darstellung eines einzelnen Programms verdient besondere Wertschätzung. Sie hilft, relativ schnell weitere Informationsquellen zu einzelnen Programmen zu finden.

Unter den vier sinnvollen Hauptkriterien

— branchenneutrale Programme für Rechnungswesen und Verwaltung,

— Branchenpakete für Rechnungswesen und Verwaltung,

— Programme für Technik und Wissenschaft,

— Systemsoftware

findet sich jeweils eine sinnvolle Feinuntergliederung. Besonders anerkennenswert ist, daß Programme, die mehreren Gliederungskriterien untergeordnet werden können, auch mehreren Gliederungskriterien zugeordnet wurden. Jeder, der einmal versucht hat die (meistens recht eigenwilligen) Zuordnungen eines Verfassers nachzuvollziehen, weiß das zu schätzen.

Zwei Zusammenstellungen der Programme, die sich für bestimmte Hardware oder bestimmte Betriebssysteme eignen, helfen dem Leser, der sich vorab informieren möchte, welche Hardware mit welchem Betriebssystem er wählen müßte, um ein möglichst breites Software-Angebot vorzufinden.

Andererseits, ein Buch, welches den Anspruch erhebt, einen Software-Markt transparent und vollständig abzubilden, muß — an diesem Anspruch gemessen — scheitern. Ständige Software-Neuentwicklungen lassen das Buch bereits am Erscheinungstag zu einem 'veralteten Werk' werden. So vermißt man zum Beispiel die neuen Entwicklungen im Bereich der 'work-sheet'-Programme.

Mit Wissen um diese wohl unvermeidbare Schwäche läßt sich das Buch allerdings hervorragend nutzen. Es ist insbesondere dem betrieblichen Anwender zu empfehlen, der für seine Probleme Programm-Möglichkeiten sammeln möchte. PK

Ludwigs/Poppensieker — Pyka/Surowiecki

UNIX für Einsteiger und Umsteiger

Köln: Verlagsgesellschaft R. Müller, 1983.
184 Seiten,
Kart. DM 39,—
ISBN 3-481-35211-5

Auch wenn es der Titel suggerieren mag: Es ist kein Buch für Diagonalleser, die auf die Schnelle einen Eindruck von UNIX bekommen wollen, sondern es ist ein Arbeitsbuch. Die Autoren gehen davon aus, daß der Leser Zugang zu einem UNIX-fähigen Rechnersystem hat, um die Beispiele aus dem Buch auszuprobieren.

Zunächst werden File-Handling und Kommando-Aufbau verständlich behandelt. Die (so die Autoren) 'weniger Glücklichen', die keine praktischen Erprobungsmöglichkeiten haben, werden oft rätseln müssen, was bei den beispielhaften Bildschirmdialogen vom Rechner stammt, und was Eingaben sind. Verschiedene Schriftarten und ein angedeutetes 'Carriage-Return' wären hier hilfreich.

Ausführlich widmen sich die Autoren dem zeilenorientierten Editor ED, der nicht nur namentlich dem CP/M-Editor ähnlich ist. Wer auf ED angewiesen ist, wird für diesen Abschnitt dankbar sein, denn nur vollendete Beherrschung beispielsweise der String-Suchfunktionen erlöst einen von ewigem Zeilen- und Zeichenzählen.

Auch wenn die Programmiersprache 'C' eng mit UNIX verknüpft ist, die 20seitige Kurzeinführung dürfte die Mehrzahl der Leser durch ihre Informationsfülle überfordern. Wenige einfache, aber genau erklärte Beispiele wären sinnvoller gewesen, zumal ein 'C'-Kursus in einer UNIX-Einführung kein Muß darstellt.

Ein knapp gehaltener, aber hinreichend informativer Überblick über Prozesse, die unter UNIX ablaufen können, und über Textverarbeitung mit UNIX bilden einen weiteren Schwerpunkt. Mit einer gut geordneten Zusammenfassung aller UNIX-Befehle (mit Erklärung) schließt das Buch.

Keine Erwähnung finden Hardware-Voraussetzungen für den UNIX-Betrieb, auch wird nicht auf verfügbare Anwenderprogramme hingewiesen.

Einsteiger im Sinne von Anfängern werden es nicht leicht haben, sich ohne Zugang zu einem UNIX-fähigen Rechner durch das Buch zu arbeiten. Hierbei muß auch erwähnt werden, daß es im gesamten Buch nicht eine einzige Illustration zur Veranschaulichung des dargebotenen Stoffs gibt.

Empfehlenswert ist es wohl in erster Linie für erfahrene Anwender, die auf UNIX 'umsteigen' wollen und mehr als ein Handbuch brauchen. AU

Wolfgang Maaß

Software-Schnellkurs dBaseII

Ein Lern- und Nachschlagewerk
Haar bei München, 1984
Verlag Markt und Technik
104 Seiten, kart.
DM 37,—
ISBN 3-922120-58-X

'Natürlich können Sie dieses Manual auch durchlesen. Aber außer einem flüchtigen Überblick gewinnen Sie nicht viel', verrät der Autor auf Seite 1. Diesen Eindruck gewinnt auch der Leser, den Untertitel 'Lernwerk' führt dieses Buch also zu unrecht.

Der Autor empfiehlt es aber als Nachschlagewerk für das erfolgreiche Datenverarbeitungsprogramm dBaseII. 'Sinnvoller ist es, für einen praktischen Fall aus dem Inhaltsverzeichnis die Funktionen herauszusuchen, die Sie brauchen — sonst beschäftigen Sie sich mit Dingen, die Sie vielleicht nie brauchen.'

Eine genauere Betrachtung zeigt aber, daß das Buch der Mächtigkeit von dBaseII nicht gerecht wird. Die meisten Funktionen und Befehle tauchen im Buch gar nicht auf. Daß dBaseII auch eine komfortable Programmiersprache ist, wird dem Leser verschwiegen.

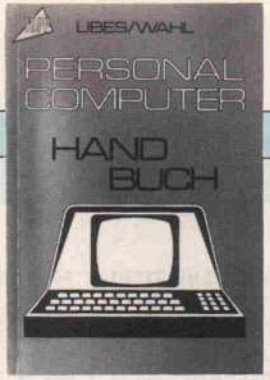
Dem Autor ist anscheinend nicht recht klar, welche weitreichenden Möglichkeiten dBaseII

zur Verknüpfung mit anderen Programmen bietet, so behauptet er, daß es 'absolut notwendig' sei, Felder der Datei mit einem Komma abzuschließen, wenn sie im Anschluß mit MAILMERGE verarbeitet werden soll. Das ist nicht so! Der COPY TO Befehl erlaubt eine Umwandlung von dBase-Dateien in Textdateien, bei denen die Felder durch Kommata oder auch beliebige andere 'Delimiter' begrenzt werden.

Das, was der Anwender und Lernende wichtiger braucht, eine Übersicht über die 'Werkzeuge' des Systems, die Befehle, findet er in diesem Buch leider nicht. Dafür auf Seite 3 einen weiteren Hinweis des Autors: 'Wenn Sie schon alles wissen, brauchen Sie dieses Manual auch nicht.'

Ein 'Lern- und Nachschlagewerk'. Für wen?

AB



S. Libes/Wahl

Personal-Computer Handbuch

Stuttgart, 1983
Frech-Verlag
184 Seiten,
kart. DM 19,80
ISBN 3-7724-0506-1

In der Einführung berichten die Autoren von Robert, einem Hobbyisten, der sich innerhalb von zwei Jahren durch die Lektüre von Computerbüchern, Bausätzen und der Mitgliedschaft in einem Computer-Club zu einem kenntnisreichen Mikrocomputer-Amateur herangebildet hat. Dem Leser trägt dieses Buch das gleiche an. Es macht einen Streifzug durch die Welt der Mikrocomputer: von Binär-Arithmetik bis zur digitalen Logik, vom Flip-Flop zur CPU und vom Assembler bis zum BASIC-Interpreter. Nirgends verweilt es lange, zeigt die grundlegenden Techniken und bietet den 'sanften Einstieg in die faszinierende Welt der Computertechnik', den der Klappentext verspricht. Es beschreibt die Peripheriegeräte vom Teletype bis zum Matrixdrucker, vom CRT-Terminal bis zum Telefon-Modem.

Zur Vertiefung seines Wissens muß der Anfänger zwar noch andere Wissensquellen anzapfen, aber dieses Buch könnte die Lektüre gewesen sein, die Robert zu seinem Hobby geführt hat — die 'Einstiegsdroge'. Was in diesem Buch beschrieben wird, ist die Grundlage und das Handwerkszeug für alle weiteren Bemühungen des Computer-Amateurs.

AB

c't-Platinen

c't-Platinen bestehen aus Epoxid-Glashartgewebe, sind fertig gebohrt und mit Lötstopplack versehen bzw. verzinnt. Die Bestellnummer bezieht sich auf den Beitrag, in dem das betreffende c't-Projekt vorgestellt wurde. Sie setzt sich zusammen aus Jahrgang, Heftnummer und Seitennummer. Die zusätzlichen Buchstaben bedeuten: 'd' — doppelseitig, 'B' — Bestückungsaufdruck, 'E' — elektronisch geprüft.

Nr.	Projekt	Format	Preis
831241dBE	Terminal A (ohne Tastatur)	ca. 84 x 234 mm	59 DM
831242dBE	Terminal B (mit Tastatur)	Doppel-Europa	75 DM
831262	Universelles Netzteil	Europa	14 DM
840147dBE	c't 86, CPU-Karte	Europa	85 DM
840148dBE	c't 86, RAM-Karte 256 KByte	Europa	88 DM
840149dBE	c't 86, I/O-Karte	Europa	69 DM
840288dBE	c't 86, Floppy-Interface	Europa	65 DM
840150d	Busplatine (96pol., 10 Steckplätze)	84 x 208 mm	49 DM
840184d	CEPAC-80 mit Wrap-Feld	Europa	69 DM
840187d	CEPAC-80 ohne Wrap-Feld	ca. 86 x 100 mm	49 DM
840242B	Centronics/V24-Interface für Olympia COMPACT	80 x 136 mm	15 DM
840252B	c't-Sprachsynthesizer	100 x 117 mm	21 DM
840352dB	CEPAC-65, Version A	80 x 100 mm	27 DM
840354dB	CEPAC-65, Version B	Europa	52 DM
840496dB	PIO-Centronics-Interface für ZX 81 (nicht durchkontaktiert)	Europa	20 DM

So können Sie bestellen: Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck oder einen von Ihrer Bank quittierten Einzahlungsbeleg über die Bestellsumme zuzüglich 3 DM (für Porto und Verpackung) bei. Bei Bestellung aus dem Ausland muß stets eine Überweisung in DM erfolgen. Die Überweisung und Ihre Bestellung richten Sie bitte an:

c't-Versand, Verlag Heinz Heise GmbH
Bissendorfer Straße 8, 3000 Hannover 61
Konto-Nr. 9305-308, Postscheckamt Hannover

Der Preishammer

Wir haben optimiert und weggelassen, was nicht unbedingt am Anfang benötigt wird.

Aber Sie haben dennoch ein optimales Gerät! Der bewährte NB-Computerbausatz mit 1 Laufwerk, Controler, Netzteil, Tastatur ..... **1650,— DM**
als Fertiggerät ..... **1850,— DM**

EPROM-Programmiergerät

Programmiert: 2708, 2716, 2732/2532, 2764, 27128 mit Zusatz auch: 8748, 8749, 87555 usw.

Bausatz ..... **175,— DM**
Zusatz ..... **90,— DM**
Fertiggerät ..... **240,— DM**
Zusatz ..... **110,— DM**

Für Apple und kompatible Rechner

Floppy-Controler 5¼" für Industrielaufwerke, z. B. BASF 6106, Shugart usw. und Originallaufwerke

Bausatz ..... **195,— DM**
Fertiggerät ..... **280,— DM**

Preh-Commander-Keyboard

AK 87 mit Gehäuse, Anschlußkabel und separatem 10er-Block, deutscher Tastensatz ..... **350,— DM**

Festplattenstation 10 MB brutto, mit sämtlicher Hard- und Software für Anschluß an Apple ..... **6000,— DM**

Sämtliche Preise inkl. MwSt.

KÜHN ELEKTRONIK
2909 Bösel · Postfach 67 · Telefon 0 44 94/15 64

DER NEUE SOFT- UND HARDWAREANBIETER! dBASE II 2.4 dt. Man. DM 1875,— alle Systeme, **MULTIPLAN** Tab-Kalk. DM 779,— alle Systeme, **ENDLOSPAPIER** 2000/240 x 12 längsp/w. DM 39,—, **ARCHIVBOXEN** f. 10 5¼"-Disk. St. DM 5,80, 10 St. DM 53,— / **STAUBSCHUTZHAUBEN** SIRIUS/IBM. Alle Preise inkl. MwSt. / Preisliste anfordern. Thomas Altmann, Computer und Zubehör, Ulrichstr. 33, 7060 Schorndorf, T. 0 71 81/7 41 20.

SUCHE 5¼"-LW BASF 6106, EXTENDED-BASIC für AIM 65. W. REICHEL, Kalchreuther Str. 98, 8500 NBG.

GOING FORTH! Endlich auch f. **SHARP** K/A/700!! **FIG-FORTH** + EDITOR + ASSEMBLER + PRINTER-ROUT!! **NUR** 80,— DM VK/NN. MAEDING. Plöner Str. 48, 2400 Lübeck.

Farbbandkassetten: Epson MX, RX, FX-80: 17,50; Itoh/NEC: 18,50; Epson MX-100: 33,— MX-100 Refill: 16,—; Reinigungsset 5,25": 79,—; Maxell-Disketten, preiswert und gut. J. Strenger Diskettenversand, Bayernstr. 15, 5628 Heiligenhaus, Tel.: 0 20 56/64 18.

Texas TI 99/4A mit Modulator, Netzteil, Cass-Rek.-Kabel, ca. 75 Programme, neuwertig, nur 350,—, Joysticks für TI 60,—, Cassettenrecorderkabel TI 99/4A für 30,—, weiteres Zubehör auf Anfrage. Box, RS232, 32 K usw. Tel. 0 21 34/9 66 80, abends.

Deutsche Software für ZX SPECTRUM von Friedr. Neuper, Leuchtenberger Str. 1, 8473 Pfreimd. Kostenloses Informationsmaterial anfordern!!!

MZ-80 A/B/K/700 Betriebssystemlistings in Deutsch, über 4000 Seiten geballte Information. Info anfordern. **sds,** Mainzer Str. 47, 5568 Daun.

ZX 81 16 K Speicher **stat.,** Einbau **in** ZX, fertig 100,— DM. N. Plack, Webergasse 5, 8520 Erlangen.

MATRIXDRUCKER NEU. Tel. 0 89/5 23 34 28.

SHARP MZ-700 SOFTWARE, Gratisliste 06 11/56 93 85.

COMPUTER-JOURNALIST für Video-Produktion gesucht. Tel. 0 52 23/8 56 25.

MZ 80 A ★ über 200 Programme ★ Spiele + kleine kommerzielle Lösungen ★ Info gegen Rückporto von K.-H. Boht, 2878 Aumühle.

Selbstklebende Tabellier-Etiketten, 22 Größen, Verkauf auch in Kleinmengen. Laufend Sonderangebote. Preisliste mit Mustern gratis bei: ULRICH KORELL ETIKETTEN-VERSAND, Postf. 13 54, 5275 Bergneustadt.

GENI I '83 mit HR-Graphik + Software DM 850,—, PET 2001 + BASIC-Lehrg. DM 490,—. T. 0 71 42/5 33 85.

Verk. **ANRUFBEANTWORTER, DRAHTLOSTELEFON,** versch. **USA-Telefone, Telefoncomputer** mit 60 Speichern. WINNER, 8000 Würzburg, Höchbergerstr. 62, Tel. 09 31/41 11 79.

STECKER für Computer liefern wir ab Lager sowie sämtliche Kabelverbindungen nach Ihren Angaben. Preisliste gegen 2,— DM in Briefmarken. COMPUTERSYSTEME NIEDERGESAESS, GOEBENSTR. 26, 6200 WIESBADEN, Tel. 0 61 21/4 59 19.

APPLE comp. Hdl-Liste Tagestiefstpreise Rückgaberecht 10 T. GENERALIMPORTEUR STREIL, Mommsenstr. 3, 4006 Erkrath 2, Tel. 0 21 04/4 30 79.

V 24- OVER 'CENTRONICS' UMSCHALTER gestattet den Anschluß mehrerer Geräte an einen Computer. INFO: Fa. Dipl.-Ing. Fricke, Wattstr. 30, 2400 Lübeck, Tel. 04 51/60 47 49 oder 60 10 27.

ACHTUNG! PREISHITS! Drucker KDC FT 5001 100Z/SEK DM 1020,— inkl., Interface-Karte m. Kabel DM 189,— inkl., TEAK-FLOPPY FD 55 B DM 740,— inkl., TEAC-FLOPPY FD 55 F DM 855,— inkl. Hardwareanpassung a. Systeme, Reparaturservice. Fordern Sie kostenlos INFO. M. MODERSITZKI, COMPUTER-SERVICE, BETTIKUMER GRUND 2A, 4040 NEUSS 21, Tel. 0 21 07/54 79 u. 6 02 69.

PREISHIT! TEAC FLOPPY-DISK LAUFWERKE: 1* FD-55A — SS,DD 40Tr, 48Tpi 250 KB 639,— DM. 2* FD-55B-DS,DD 40 + 40Tr, 48Tpi 500 KB 739,— DM. 3* FD-55E-SS,DD 80Tr, 96Tpi 500 KB 749,— DM. 4* FD-55F-DS,DD 80 + 80Tr, 96Tpi 1 MB 869,— DM. 5* FD-55G-DS,DD 80 + 80Tr, 96Tpi 1, 6 MB 998,— DM. Incl. MwSt. Vers. Kosten 9,90 nur p. NN o. V.Kasse. Z. LUKAVSKY, TH.-WIMMER-STR. 33, 8058 ERDING.

An dieser Stelle könnte Ihre private oder gewerbliche Kleinanzeige stehen. Exakt im gleichen Format: 8 Zeilen à 45 Anschläge einschl. Satzzeichen und Wortzwischenräumen. Als priv. Hobby-Elektroniker müßten Sie dann zwar 31,92 DM, als Gewerbetreibender 52,90 DM Anzeigenkosten begleichen, doch dafür würde Ihr Angebot auch garantiert beachtet. Wie Sie sehen.

Verk. IBM-KK-Drucker + Traktor + Interface, 600 DM. Olympia-KK-Schreibmasch. (8080-CPU + 16K-RAM), 17z/s + Interface, 1250 DM / Tastatur (Keytronic), parallel, 250 DM. Tel. 05 21/87 41 15 ab 19 Uhr.

VC-20 zu verkaufen. Diverse Software (über 40 Prgr.) 40/80-Zeichenkarte, 8k-Erweiterung, Steckplatzerweiterung, Begleitbücher. Alles zusammen für 800,— DM. Kaufpreis über 1000,— DM. Auch Tausch gegen C-64. Kein Teil älter als 2 Monate. Von Privat. Kai Holst, Hohenwischerstr. 67, 2101 Hamburg 96.

Suche defekten Spectrum. Tel. 0 23 03/1 33 45.

NEC-PC 8800 zu verkaufen. CP/M 2.2 BASIC 40 KB 2 · 1MB Fl. 8" H-Gr. 400.640P Te. Nr. 0 23 23/3 10 15.

KEINE ÜBERLADUNG DES HX-20 durch Ladegerät N-PLUS, DM 129,—. INFO: Fa. Dipl.-Ing. Fricke, Wattstr. 30, 24 Lübeck, Tel. 04 51/60 47 49, BZN 60 10 27.

Zahle 200 DM für **ZX-SPECTRUM.** Tel. 0 65 87/70 07.

CP/M CROSS ASSEMBLER. Ideal für Einplatinencomputer Entwicklung 6809 65XX 6800 je DM 249. D. CORSON, Röntgenstr. 13, 5900 Siegen 0 27/8 88 15.

4116 33,— / 32 ST. 28,— / 24 ST. 21,— / 16 ST. 12,— / 8 ST. 2,— / ST. + 5,— Porto/Verpackung. HERGENHAHN, Isenburgring 14, 6050 Offenbach.

VERKAUFE SHUGART 8" DH DD, Terminal V24, Terminalgehäuse 8,5" Schirm, Tastatur, Videokarte DM 350, 8" Laufwerke DM 300, DIHACO Computer bzw. INTERTECHNIC, PHILIPS DIASTEUERUNG 30 DM. 8 Kanal Locher u. Leser DM 150. Tel. 0 26 03/47 48.

ZX Spectrum: 3-Kanal-Soundgenerator **AY-3-8910** (c't 12/83). Mit 2 Ports, **Centronics**-Kabel und -Software nur 169,— DM. **EPROM-Burner** (2716) m. **3 I/O-Ports** nur 129,— DM. M. Lücker, 02 41/50 31 62.

Staubschutzhauben aus reißfestem, beschichteten Textilgewebe in neutraler Farbe für Computer, Drucker usw. nach Ihren Maßangaben. Preise: z. B. für **COMMODORE 64 nur 15 DM** incl. MWSt., Porto, Verpackung bei Vorausscheck, sonst plus NN-Gebühr. Evtl. Info anfordern bei Fa. Fritz Jung KG, Bahnhofstr. 14, 8620 Lichtenfels.

MAXI-QUALITÄT zum Mini-Preis **C-64 EPROM-Modul Platine** 8 KB für den $8000-$9FFF-Bereich nur DM 30. Tel. 0 23 33/8 02 02 nach 17 Uhr.

HP-85, 32K, HP-IB(IEC), PRINTER-PLOT. ROM,ADV. BASIC ROM, div. Cass., Papier, etc., sehr gepflegt, VB 5300 DM. Tel. 0 23 30/62 38 33.

Acorn/B — (BBCIB) — Computer zu verk. Tel. 06 51/7 62 84.

64-FORTH. Nun gibt es diese schnelle struktur. SPRACHE auch f. d. **C-64.FORTH** ist bel. erweiterbar und damit für Graphik u. Soundprogr. gut geeignet. **FORTH** a. Cass. DM 49,—. Handbuch DM 25,—. INFO D. Luda, Staudingerstr. 65, 8000 München 83 ★ 64-FORTH ★ 64-FORTH ★ 64-FORTH

Spectrum im MC-Programm "abgestürzt"? Reset ohne Programmverlust-Bananl.-Info gegen adr. Freiumschl. Postf. 25 32, 3300 Braunschweig.

Haben Sie vielleicht einen der folgenden Computer: AIM-65, VC20/64, TRS-80/VG, ZX 81, APPLE. CP/M-Computer, . . . ? Dann sollten Sie unbedingt unser kostenloses Info "CT5" anfordern: **D. Kiesenberg, PF 5 79, 4600 Dortmund 1.**

DURCHKONTAKTIEREN ohne Spezialwerkzeug mit Kupferhohlnieten: DM/1000 St. bei Außendurchmesser 1 mm 24,— ★ 1,2 mm 26,— ★ 1,5 mm 27,— ★ 1,8 mm 28,— plus Versandkosten per Nachnahme. Elmar WIENECKE, Wasserstraße 18, 4973 Vlotho 1, Tel. 0 57 33/58 01.

BESTELLEN SIE NOCH HEUTE, gegen adressierten Freiumschlag UNSERE PERIPHERIE-LISTE! Inhalt: Interface parallel—seriell, Light Pen, Speichererw., Schnittstellen-Anpassung für Drucker & Floppy und vieles mehr. Für Apple, Com., IBM, Sinc., Tandy, TI. etc. . . . ! ★ KAUFEN & BEZAHLEN SIE NUR, WAS SIE SUCHEN ★ Unsere Schaltungsmappe ist kein Buch zum mehrfachen Preis. Sie ist genau darauf abgestimmt, was sie benötigen! F. Warkus — Postf. 31 50 — 4920 Lemgo 1.

● **COMUTER-CASSETTEN** im 10er Pack, BASF-Band-LHD, mitBox, Etikett u. Einleger, C 10 nur 15 DM, C 20 16 DM, C 30 17 DM. **BRANDNEU von TDK:** PC-10 im 10er Pack 29 DM, PC-15 32 DM ● **CASSETTEN-AUFKLEBER** auf Lochstreifen, 100 St. 5 DM, 120 St. auf A4-Druckbögen 7 DM, **VIDEO-CASSETTEN** Domiphon VHS E-180 17,50 DM, Christomenia-Cassettenstudio, 3584 Zwesten, Postf., Tel. 0 56 26/2 81. Versand ab 20 DM.

VC64 VC64 VC64 VC64 VC64 VC64. SUPERSPIELE und **PROGRAMME** — Tausch oder gegen Unkostenbeitrag. Gleich Info anfordern! Es lohnt sich! D. Walthemathe, H.-Heine-Str. 4B, 3260 Rinteln 8, Tel. 0 57 51/56 07.

Adreßverwaltung unter CP/M 80 u. CP/M 86, Zugriff über 6 Schlüssel in sec.schnell. Edik. Druck Erstellen einer WS Datei für Seriendruck. Als Telefonregister verwendbar, über weitere Programme Info anfordern. Preis 260 DM. Telefon 07 11/7 49 54, Eugen Mayer, Mozartstr. 37, 7141 Freiberg.

SPECTRUM Supersoftware Info 80 Pf. Dipl.-Ing. Gerd Verse, Grüner Weg 45, 4650 Gelsenkirchen.

LASER-II incl. Forth, Basic + CP/M 1334,— 16k, Z80 Ctr-Card à 125,—, Joystick 35,—, 80 Zeichen 218. Weilacher, Stockmannstr. 44, 8000 Mchn 71, 0 89/79 46 30.

COMMODORE 64, **alle** Anwender u. Spielprogramme, z. B. 10 Disketten mit Spielen DM 150,—. Tel.: 02 01/60 27 06.

C-64 Interface IEEE-Bus für alle CBM-Peripherie-Geräte wie z. B. Floppy 8050, Drucker u. Meßgeräte. Der volle Speicher bleibt erhalten, voll kompatibel zu den Maschinenprogrammen. Einführungspreis 248 DM. Franke, Postfach 10 12 23, 4300 Essen. Tel. 02 01/73 77 64.

FLOPPY SHUGART SA 400, wie in c't beschrieben, VB 350,—. Tel. 0 91 22/1 55 04.

SPEEDY 64, der schnelle C-64 BASIC-Compiler nur DM 75,—. Kompaktor DM 20,—. Disk-Copy DM 15. **VC-20 BASIC-Compiler** nur DM 50,—. Info 80 Pf. Klaus Raczek, Wickrathberger-12, 5140 Erkelenz.

6522-Universalinterface für **Apple II** nach MC 3/83, durchkont. Leiterpl. m. Lötstop + Pos.-Druck **35,— DM** / Bausatz kompl. **128,— DM.** Fertigplatine geprüft **158,— DM** + Vers. p. NN. Fa. digitan-analog E. Balter, Hauptstr. 96, 8752 Krombach, Tel. 0 60 24/98 45.

ZX 81 Zubehör für den versierten Bastler: Platinen, Bausätze, Schaltpläne (auch einzeln!) ★ universeller Epromer ★ 16/64 Kdyn RAM ★ Z-80 Assembler-Editor-Debugger ROM ca. 205. Info für 1,80 in Bfm. Veith, Speidelweg 9, 7000 Stuttgart 61.

Das Handbuch zum elrad-COBOLD-Computer!

Christian Persson
6502/65C02
MASCHINENSPRACHE
Programmieren ohne Grenzen

1983, ca. 250 Seiten mit vielen Abbildungen, Großformat DIN A4 quer. DM 48,—

Drei Bücher in einem!

Programmierkursus: Eine 'locker geschriebene', praxisnahe Einführung *in die uC-Technik und -Programmierung*, die keine Vorkenntnisse verlangt. Die umfassende Anleitung vom ersten Tastendruck bis zum Entwurf komplexer Systemprogramme. *Mit dem COBOLD-Computer steht ein komfortables Trainingssystem zum Selbstunterricht zur Verfügung, das nach der 'Lehrzeit' seinen Wert behält!*

Programmsammlung: Leistungsfähige Standard-Routinen, wie sie jeder 6502-Anwender oft braucht — *zum Teil in sich abgeschlossene Bestandteile des 4-KByte-Betriebssystems: Rechenprogramme, Such- und Sortierprogramme, Karteiverwaltung, Peripherieansteuerung, Serielle Datenübertragung, schnelle Kassettenrecorder-Software (4800 Baud), Multiplex-Display, Tastaturabfrage, Codieren/Decodieren und vieles mehr. Ein Nachschlagewerk für den Software-Entwickler.*

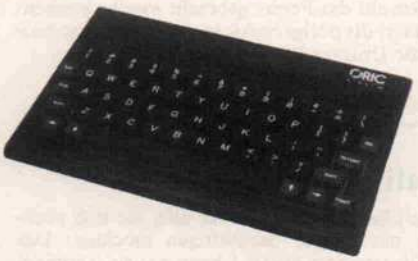
COBOLD-Dokumentation: Die unentbehrliche Arbeitsgrundlage für den COBOLD-Anwender. Beschreibt Hardware und Software in allen Details: *Monitor-, Editor-, Texteditor-Befehle, Assembler, Disassembler, Kassettenaufnahme, Integrieren externer Programme, Terminal-, Drucker-, TTY-Anschluß und vieles mehr. Die große Vielseitigkeit des COBOLD-Computers wird nutzbar gemacht.*

Versandbedingungen: Die Lieferung erfolgt per Nachnahme (plus DM 5,00 Versandkosten) oder gegen Verrechnungsscheck (plus DM 3,00 Versandkosten).

Verlag Heinz Heise GmbH
Postfach 27 46
3000 Hannover 1

Der neue ORIC ATMOS spricht für sich.

Bei dieser Qualität und dem Preis von **nur 748,— DM** führt kein Weg am neuen ORIC vorbei!



Ihr ORIC-Spezialist für Norddeutschland hält den ATMOS zur Vorführung **mit Sprachsynthesizer** etc. für Sie bereit.

(Auch schriftl. Händleranfragen erwünscht.)

UTAW Electronic
Laser- und Computertechnic
Hagenstraße 31 — 3000 Hannover 1
Telefon (05 11) 34 10 38

Zubehör für VC 20 + C 64

Die hier angebotenen Zubehörteile werden anschlußfertig geliefert. Es ist kein Löten mehr erforderlich. Wir halten aber auch Bausätze bereit. Fragen Sie nach unserem Preis! Alle Artikel werden mit Anleitung geliefert.

VC-Modulplatine f. 5 Steckmodule	DM	99,80
VC-Modulplatine f. 3 Steckmodule	DM	72,10
Recorderinterface f. C.-Recorder	DM	35,70
EPROM-Programmierer 2716/32/64/128	DM	245,95
VC 20-EPROM-Karte (2716/2732)	DM	52,80
VC 20-64 KByte RAM, voll schaltbar	DM	239,00
VC 20-40/80-Zeichenkarte	DM	225,00
C 64-80-Zeichenkarte, Steckmodul	DM	279,00
C 64-EPROM-Karte (2716/2732/2764)	DM	55,00
Textverarbeitung (VC 20/64)	DM	139,00
Assembler (VC 20/64	DM	29,50
Vokabellehrer 64 (C 64)	DM	39,80
GP 100 VC (VC 20/64)	DM	649,00
EPSON RX 80 VC (VC 20/64)	DM	1398,00
COMMODORE 64/VC 20-BASIC-Kurs	DM	49,00
Joystick für COMMODORE-Computer	DM	44,85
COMMODORE 64 mit Rekorderinterf.	DM	738,50

Außerdem führen wir VC 20/64-Literatur und VIEL Software. Bitte fordern Sie unser VC 20/64-Info gegen einen Freiumschlag an. Der Versand erfolgt per Nachnahme zzgl. Porto + Verpackung.

VE-Computer-Systeme GmbH
Hard- und Software-Entwicklungen
Goethestraße 46, 4390 Gladbeck
Telefon (0 20 43) 6 37 03 ab **14 Uhr.**
Vorkasse: + DM 6,00 auf PSK. 39 31-437 ESSEN

DEDUGGER-TESTER

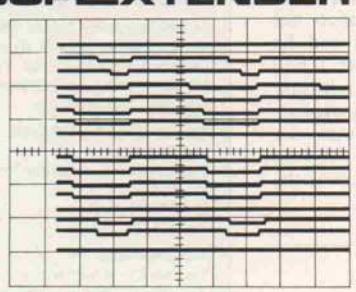
- Für **Z80, 8085, 6502, 6800, 6802, 6809** usw. **Prozessoren**
  Ein preisgünstiges Gerät zum Testen Ihrer Assembler-Programme mit der dazugehörenden Hardware
- Leichte Bedienung durch Tasten mit LED
- LCD-Anzeige des Adreß- und Datenbusses
- Einstellbare Triggerbedingungen für Single-Step Mode oder Latch — Mode für Realtime-Test
- Einfacher Anschluß über CPU-Sockel

Preis inkl. MwSt. ..... **DM 956,50**

Entwicklungsbüro F. Krickl
Schauinslandweg 27, 7730 VS-Schwenningen
Telefon: (0 77 20) 6 12 33

SCOPEXTENDER

Fertiggerät DM 137,—
Bausatz DM 69,—
Netzteil DM 35,—
Gehäuse DM 20,—



W.GROTJAN
ELEKTRONIK-VERSAND
NEUE STR. 1
3305 VELTHEIM / OHE

16-Kanal Realtime-Darstellung auf jedem Oszilloscop

volle TTL-LS-Arbeitsgeschwindigkeit

c't

magazin für computer technik

7/84 — Anzeigenschluß am 8. Mai 1984

Firmenverzeichnis zum Anzeigenteil

afu-electronic, Meschede 100
Aumann, Strassenhaus 100

BSP Krug, Regensburg 24
bst computronic, Frankfurt 91

CE Computer, Schwerte 99
Christiani, Konstanz 11
Conitec, Darmstadt 100

Data Becker, Düsseldorf 33

Ebeco, Köln 49
ESH-Handelsagentur, Hirtscheid ... 91
Elektronika Saar, Saarbrücken 97
Elektronikladen, Detmold 100
eps, Großrinderfeld 27

Frech-Verlag, Stuttgart 51
Frölje, Oldenburg 17

Grotjan, Veltheim 107

HIB, Nürnberg 101
Hösch, Düsseldorf 100

Instrumatic Electronic, Gräfelfing .. 97

Krickl, VS-Schwenningen 107
Kühn, Bösel 105

LAMPSON & ZERBE, Büttelborn ... 5
Linde, Schwäbisch-Hall 24
LOGITEK, Berlin 101
Luther-Verlag, Sprendlingen 103

MARFLOW, Hannover 79, 81
Michels & Kleberhoff, Wuppertal .. 112
Microcomputersysteme, Achim . 12, 13
Mikrocomputerladen, Berlin 99
mirwald, Unterhaching 99
MSE electronic, Düsseldorf 19
Münzenloher, Holzkirchen 91

Neuhaus, Dr., Hamburg 51
NEWMAN Computer-Versand, Schenefeld 97

profisoft, Osnabrück 65

RATEV, Ratingen 100
RB Elektronik, Eitorf 19
RK Computer, Korbach 24
Rösch, Braunschweig 101
ROOS elektronik, Kleve 101

SE-Spezial Electronic, Bückeburg .. 51
Siemens, Stuttgart 91
SYNELEC, München 15
Syntax, Rastatt 101
Syscom, Winkler, Heidelberg 25
Systech, Braunschweig 21

Schlüter, Castrop-Rauxel 25

STA, Heidelberg 24, 103

Technitron, München 9
Teepe, Wellrod 44
te-wi-Verlag, München 24, 49
Thoma, Illertissen 23

UTAW, Hannover 107

VE-Computer, Gladbeck 107
Völzke, Pullach 100

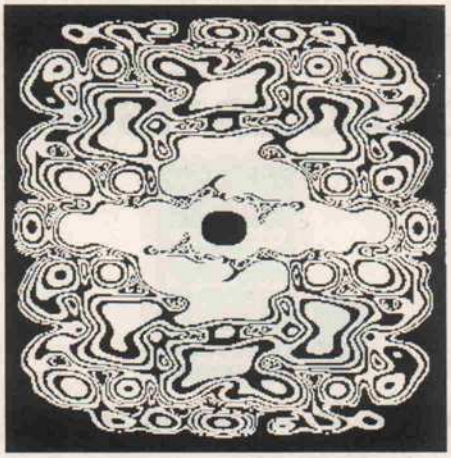
WCS-WEMPER, Aumühle 101
Weber, München 101
Wersi, Halsenbach 25
Wittke + Riegraf, Filderstadt 100

In der nächsten c't

unter anderem

Schwerpunkt Grafik: Mathematik visualisiert

Die kunstvollen Ornamente aus dem Computer sind keine Zufallsprodukte. Der Rechner erweist sich als hervorragendes Mittel der Visualisierung mathematischer Zusammenhänge, wenn man ihn entsprechend zu programmieren versteht. Ganz 'nebenbei' sind die entstehenden Bilder auch noch schön. Der Physiker und Science-fiction-Autor Prof. Dr. Herbert W. Franke hat sich in den vergangenen Jahren als Experte für Computergrafik einen Namen gemacht. Er berichtet in c't über Arbeitsergebnisse und -verfahren.



Fouriertransformation eines Buchstabenzeichens. Bildautoren: Herbert W. Franke und Horst Helbig.

Apple-Grafik

Einen Großteil seiner weiten Verbreitung verdankt der Apple seinen für 'damalige' Zeiten richtungweisenden Grafikfähigkeiten. Unsere Beitragsreihe beginnt mit einer allgemeinen Funktionsbeschreibung der Apple-Grafik, die auch den Computerfreunden, die keinen Apple besitzen, das Verständnis des vorgestellten Programmpaketes ermöglicht. Wesentliches Ziel dieser Beiträge ist es, darzustellen, wie man bewegte Grafiken im Hinblick auf Geschwindigkeit optimiert.

Grafik drucken

'Grafikfähig' ist heute fast jeder neu auf dem Markt eingeführte Drucker. Bei den meisten bedeutet das, daß im 'Bitmuster-Modus' einzelne Punkte auf das Papier gebracht werden können. c't liefert die nötige Software und das Know-how für die Drucker-Einzelpunktgrafik.

Grafikkarte GRIP-1

Ein c't-Selbstbauprojekt für alle, die sich praktisch mit Grafik beschäftigen möchten: Das Grafik-Interface GRIP-1 kann wie ein Terminal entweder über den ECB-Bus oder über eine serielle Schnittstelle angesteuert werden. Die Auflösung beträgt 768 x 280 Bildpunkte, so daß handelsübliche Monitore verwendet werden können; Text läßt sich natürlich auch darstellen. Ein eigener Z 80-Prozessor ist fürs Vektorenzeichnen und für die Bildschirm- und Schnittstellenverwaltung zuständig. An die Europakarte können zusätzlich noch Tastatur, Drucker, Lichtgriffel und Lautsprecher angeschlossen werden.

Brother CE50 als Typenterminal

Eine gute Tastatur kostet (mit Gehäuse) so um die 500 Mark, ein Low-Cost-Typenraddrucker etwa das Dreifache. Wer beides haben und dennoch nur rund 1000 Mark ausgeben will, wählt am besten eine Typenradschreibmaschine mit Computeranschluß. Ein Musterbeispiel sind die beiden kompatiblen Typenradmaschinen CE50 und CE60 von Brother. Sie sind bereits mit einer Schnittstelle zum Interface-Anschluß ausgestattet. Wir haben das Betriebsprogramm der Schreibmaschine geknackt und zeigen, wie man Tastatur und Druckwerk auch ohne zusätzliche Hardware nutzen kann.

Außerdem:

Weitere SuperTape-Routinen
Label-BASIC für Dragon
VC-20 als intelligente Tastatur
ORIC-ROM-Adressen
u.v.a.m.

Heft 6/84 (Mai/Juni) erscheint am 17. Mai 1984
Änderungen vorbehalten

Das bringt elrad

elrad 4/84 — jetzt am Kiosk

● Bauanleitung Bühne/Studio: PA-Box ● Bauanleitung 'Elektronische Sicherung' ● Die elrad-Laborblätter: Der Operationsverstärker in Meß- und Prüfschaltungen; zahlreiche Schaltungsbeispiele ● elrad-Report: Musikmesse Frankfurt — die neuen Techniken ● Computing Today: FORTH-Simulator in ZX-BASIC

elrad 5/84 — ab 24. 4. 1984 am Kiosk

● elrad-Report: Akkus und Batterien ● Bauanleitung: Parametrischer Equalizer ● Bauanleitung: LCD-Thermometer für zwei Meßstellen ● Bauanleitung: Scheibenwischer-Intervallschalter ● Bauanleitung: Berührungs- und Annäherungsschalter

Impressum:

c't
Magazin für Computertechnik

Verlag Heinz Heise GmbH

Bissendorfer Straße 8, 3000 Hannover 61
Postanschrift: Postfach 27 46
3000 Hannover 1
Ruf (05 11) 5 35 20

technische Anfragen nur freitags 9.00—15.00 Uhr

Postscheckamt Hannover, Konto-Nr. 93 05-308
Kreissparkasse Hannover, Konto-Nr. 000-019968
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Andreas Burgwitz (stellvertretender Chefredakteur)
Dipl.-Ing. Detlef Grell

Ständige Mitarbeiter:
Dipl.-Ing. Rolf Keller
Dipl.-Ing. Eberhard Meyer
Dipl.-Chem. Holger Petersen
Dipl.-Ing. Eckart Steffens
Dipl.-Ing. Kurt Werner

Technische Assistenz: Hans-Jürgen Berndt

Abonnementsverwaltung, Bestellwesen:
Dörte Imken, Anita Kreutzer

Anzeigen:
Wolfgang Penseler (Anzeigenleiter)
Gerlinde Donner (Disposition)

Es gilt die Anzeigenpreisliste 1 vom 1. 10. 1983

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:
Verlag Heinz Heise GmbH
Postfach 27 46
3000 Hannover 1
Ruf (05 11) 5 35 20

Herstellung: Wolfgang Ulber

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Satz und Druck:
Hahn-Druckerei, Im Moore 17, 3000 Hannover 1
Ruf (05 11) 708 370

c't erscheint monatlich.
Einzelpreis DM 6,—, öS 52,—, sfr 6,—, hfl 6,80,

Jahresabonnement Inland DM 58,— inkl. MwSt. und Versandkosten. Schweiz sfr 58,— inkl. Versandkosten. Österreich öS 480,— inkl. Versandkosten. Niederlande hfl 68,— inkl. Versandkosten. Sonstige Länder 65,— DM inkl. Versandkosten.

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):
Verlagsunion Zeitschriften-Vertrieb
Postfach 57 07
D-6200 Wiesbaden
Ruf (0 61 21) 266-0

Verantwortlich:
Textteil: Christian Persson
Anzeigenteil: Wolfgang Penseler
beide Hannover, Bissendorfer Straße 8,
3000 Hannover 61

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Die geltenden gesetzlichen und postalischen Bestimmungen bei Erwerb, Errichtung und Inbetriebnahme von Sende- und Empfangseinrichtungen sind zu beachten.

Die gewerbliche Nutzung, insbesondere der Schaltpläne und gedruckten Schaltungen, ist nur mit schriftlicher Genehmigung des Herausgebers zulässig. Die Zustimmung kann an Bedingungen geknüpft sein.

Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Nachdruck nur mit Genehmigung des Verlages. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das Exclusivrecht zur Veröffentlichung. Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden.

Sämtliche Veröffentlichungen in **c't** erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Printed in Germany
© Copyright 1984 by Verlag Heinz Heise GmbH

ISSN 0724-8679

Titelidee: **c't**
Titelfoto:
Fotozentrum Hannover, Manfred Zimmermann

c't - Private Kleinanzeige

Auftragskarte

Private Kleinanzeigen je Druckzeile DM 3,99 inkl. MwSt.

Gewerbliche Kleinanzeigen je Druckzeile DM 6,61 inkl. MwSt.

Chiffregebühr DM 5,70 inkl. MwSt.

c't - Kleinanzeige Auftragskarte

Bitte veröffentlichen Sie in der nächsten erreichbaren Ausgabe nachstehenden Text:

DM 3,99	
DM 7,98	
DM 11,97	
DM 15,96	
DM 19,93	
DM 23,94	
DM 27,93	
DM 31,92	

Pro Zeile bitte jeweils 45 Buchstaben einschl. Satzzeichen und Wortzwischenräume. Wörter, die **fettgedruckt** erscheinen sollen, unterstreichen Sie bitte. Den genauen Preis inklusive Mehrwertsteuer können Sie so selbst ablesen. Soll die Anzeige unter einer Chiffre-Nummer laufen, so erhöht sich der Endpreis um DM 5,70 Chiffre-Gebühr inkl. MwSt.

Bitte umstehend Absender nicht vergessen!

c't-Kontaktkarte

Mit dieser Service-Karte können Sie

- **Informationen** zu in **c't** besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierenden oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, **c't**-Versand, Postfach 2746, 3000 Hannover 1, **ordern**.

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in **c't** _____/8__, Seite _____ erschienene

☐ Anzeige ☐ redaktionelle Besprechung

☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____

☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't-Kontaktkarte

Mit dieser Service-Karte können Sie

- **Informationen** zu in **c't** besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierenden oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, **c't**-Versand, Postfach 2746, 3000 Hannover 1, **ordern**.

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in **c't** _____/8__, Seite _____ erschienene

☐ Anzeige ☐ redaktionelle Besprechung

☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____

☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

Absender (Bitte deutlich schreiben!)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Bitte veröffentlichen Sie den umstehenden Text von ____ Zeilen zum Gesamtpreis von _____ DM in der nächsterreichbaren Ausgabe von **c't**. Den Betrag habe ich auf Ihr Konto

Postscheck Hannover,
Konto-Nr. 93 05-308;
Kreissparkasse Hannover,
Konto-Nr. 000-0 199 68

überwiesen/Scheck liegt bei.

Veröffentlichungen nur gegen Vorauskasse.

Datum Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

Antwort

Bitte mit der jeweils gültigen Postkartengebühr freimachen

magazin für computer technik

Anzeigenabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

c't - Private Kleinanzeige

Auftragskarte

Nutzen Sie diese Karte, wenn Sie etwas suchen oder anzubieten haben!

Abgesandt am

_____ 198__

Bemerkungen

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen. ▶

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

_____ 198__

an Firma _____

Bestellt/angefordert

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen. ▶

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

_____ 198__

an Firma _____

Bestellt/angefordert

Heise Software

c't-Programme

Programme auf Datenträgern aus den Rubriken 'c't-Programme', 'c't-Projekt' und 'Computer zu Hause'.

Nr.	Programm	Datenträger	Preis
S831241	MINIMON (Z 80-Monitor)	Kassette (TRS-80)	5 DM
S831241	Terminal-Betriebsprogramm	EPROM (2732)	20 DM
S831242	Terminal-Zeichensatz ZS0	EPROM (2732)	20 DM
S831243	Zeichensatz ZS1 (deutsch)	EPROM (2732)	20 DM
S831272	Textbausteinprogramm	Kassette (TRS-80)	5 DM
S831292	Master-Directory	5¼-Zoll-Floppy	15 DM
S840145	c't 86-Monitor	2 EPROMs (2732A)	69 DM
S840146	c't 86 Monitor-Assemblerlisting	39 Seiten DIN A4	6 DM
S840172	65C02-Assembler in FORTH	5¼-Zoll-Floppy (Apple)	15 DM
S840232	Energiekostenberechnung für Strom und Gas mit ZX81	Kassette	5 DM
S840324	Kfz-Kostenanalyse mit Video Genie	Kassette	5 DM
S840349	SPRITE-Editor für C 64 (3 Versionen: Floppy-, Kassettenbetrieb und schnelles Maschinensprache-Programm)	Kassette (C 64)	5 DM
S840449	Gesamtlohnsteuertabellen 1984	Kassette (ZX 81)	5 DM
S840452	Mischkulturen (Gartenplanung)	Kassette (Spectrum)	5 DM
SuperTape			
S840423	SuperTape für ZX 81 (Basisroutinen, Betriebsprogramm und Kaltstart-Lader im ZX 81-Format)	Kassette	5 DM

Programmbibliothek

Im c't-Software-Service erhalten Sie ein Sortiment besonders interessanter und leistungsfähiger Programme für verschiedene Computersysteme, das ständig erweitert wird. Allen Programmen sind ausführliche Erläuterungen, zum Teil in Handbuchform, beigefügt.

Programmbibliothek Nr. 1
(für PET 2001 (ab 8 KB), cbm 3001, TRS-80 Level II)
10 lehrreiche und unterhaltsame BASIC-Programme, u. a. Schnell-Lese-Training. Übung für das Präzisionsschreiben, Drill für das Kopfrechnen, Berechnung von Zinseszinsen, der Computer als Hellseher.
Programmkassette 19,80 DM
Handbuch (56 Seiten) allein 8,80 DM

Programmbibliothek Nr. 2
(für PET 2001 (ab 8 KB), cbm 3001, TRS-80, Level II)
10 BASIC-Programme, u. a. Drillprogramm für das Bruchrechnen, Übung für das Geschwindigkeitsschreiben, Tilgungsplan für ein Darlehen, Reaktionszeit-Test, Gedächtnis-Training, Trainingsprogramm für die Beobachtungsgabe, der Computer als Post.
Programmkassette 19,80 DM

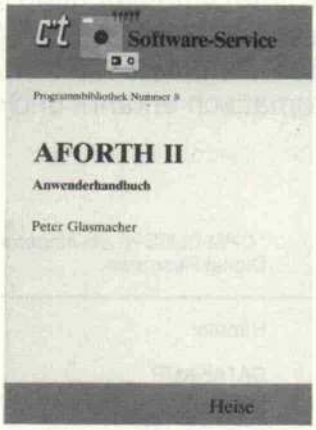
Programmbibliothek Nr. 3
RHINO
(für PET 2001 (ab 8 KB), cbm 3001, TRS-80 Level II)
Ein spannendes Spiel für intelligente Leute. Mit vielen Variationsmöglichkeiten.
Programmkassette 19,80 DM

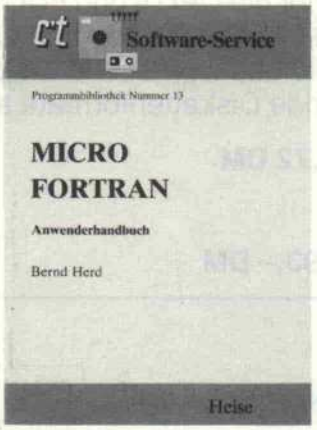
Programmbibliothek Nr. 4
Analog-Uhr/Digital-Uhr
(für PET 2001 (ab 4 KB) und cbm 3001)
Programmkassette 19,80 DM

Programmbibliothek Nr. 5
Morse-Tutor
(für PET 2001 (ab 8 KB), cbm 3001)
Übungsprogramme für das Erlernen des Morse-Codes. Die akustische Ausgabe erfolgt mit Hilfe eines anzuschließenden Radios oder Kassettenrekorders.
Programmkassette 19,80 DM

Programmbibliothek Nr. 6
PACK/UNPACK
(für PET 2001 (ab 8 KB) und cbm 3001)
Ein sehr nützliches Dienstprogramm zum Anlegen, Ändern/Ergänzen und Lesen von Dateien aus numerischen Daten.
Programmkassette 19,80 DM

Programmbibliothek Nr. 7
Interaktive Menüplanung
(Für Commodore. Mind. 32 KByte oder Diskette. Fordern Sie unseren Spezialprospekt an.)
Gesünder essen mit Computer-Hilfe: Das Programm enthält für die meisten gängigen Lebensmittel (fast 400) Informationen über Energie-, Nährstoff-, Mineralstoff- und Vitamingehalt. Es ermöglicht die Zusammenstellung von Mahlzeiten im Dialog mit dem Computer. Ideal für alle, die beruflich mit dem Erstellen von Speiseplänen zu tun haben, aber auch für den privaten Haushalt. Wahlweise Druckerausgabe.
Diskette oder Kassette 92,50 DM





Superhits für VC 20 und C 64!

Programmbibliothek Nr. 10
TEXTY
Ein großes Textverarbeitungsprogramm mit folgenden Features:
- Einlesen und Speichern von Texten auf Kassette oder Diskette
- Text erfassen mit Anzeige von Zeilen- und Spaltenposition
- Neue Zeile einfügen
- Druckerausgabe
- Kopieren von Zeilen
- Suchen von Textstücken; mit der Funktion Ersetzen kann der gefundene Textteil durch einen neuen, wahlweise kürzeren oder längeren Text ersetzt werden.
- Voll menügesteuert

Es sind zwei verschiedene Versionen mit spezieller Druckeranpassung erhältlich:
TEXTY GP für Commodore-Drucker VC 1515, VC 1541 und Seikosha GP 80, GP 100 VC
TEXTY MX für Epson MX 80
Kassette mit Handbuch 49,— DM

Programmbibliothek Nr. 11
ADRESSEN
Anschriften von Freunden, Verwandten, Vereinsmitgliedern, Kunden, Lieferanten werden verwaltet und in übersichtlicher Form angezeigt. In Zusammenhang mit TEXTY Adressenausdruck für Serienbriefe realisiert.
Kassette mit Handbuch 39,— DM

Programmbibliothek Nr. 12
KARTEIKASTEN
Dieses Programm macht alle Karteikästen überflüssig. Es erlaubt die Verwaltung beliebig großer Karteien (nur durch Speichergröße begrenzt). Möglich sind
- Anlegen einer neuen Datei
- Abspeichern auf Band oder Diskette
- Einlesen bestehender Dateien von Band oder Diskette
- Sortieren nach auszuwählenden Feldern
- Druckerausgabe mit vielen Möglichkeiten.

Für jede Kartei lassen sich beliebig viele List-Ausdrucke festlegen und ebenso wie die Datensätze auf Band oder Diskette speichern. Alle Funktionen werden über Menüs gesteuert.
Kassette 49,— DM

Bitte beachten Sie: Die Programme sind in verschiedenen Versionen für C 64 und für VC-20 mit mindestens 16 KByte RAM (Erweiterung) erhältlich. Bitte geben Sie deshalb bei der Bestellung den Rechnertyp an.

Unser Bestseller:

Programmbibliothek Nr. 8
FORTH mit 65C02-Assembler
(für Apple und Apple-kompatible Computer mit Diskettenlaufwerk)
Das Programm enthält neben einem FORTH-Compiler nach dem FORTH-79-Standard einen zeilenorientierten Editor und einen Assembler für den erweiterten Befehlssatz der CMOS-CPU R65C02. Wenn das System mit einer 80-Zeichen-Karte ausgestattet ist, steht zusätzlich ein komfortabler Screen Editor zur Verfügung.
In 64-KByte-Systemen wird FORTH in die Language-Karte geladen und belegt den Adreßbereich (H) D000...F7FF. Die Transient Program Area (TPA) beginnt bei (H) 5000, so daß für High-Resolution-Anwendungen noch eine Seite frei bleibt. Bei anderen Systemen wird FORTH ab (H) 5000 geladen. Es steht dann mehr als 10 KByte Speicherraum für Anwenderprogramme zur Verfügung — wesentlich mehr als bei herkömmlichen FORTH-Systemen.
Der Compiler wird auf einer Diskette (Format: Apple Standard) geliefert, deren Rückseite das Source Listing des Assemblers und des Editors sowie nützliche Utilities wie einen FORTH-Decompiler und einen Textformatierer enthält. Es ist geplant, nach Festlegung des FORTH-83-Standards ein Anpassungsprogramm anzubieten.
Diskette mit Handbuch 98,— DM
Zwei Disketten (single sided) mit Handbuch 113,— DM

Programmbibliothek Nr. 9
MYSTERY
(für ZX 81)
Ein Spiel, das nie langweilig wird. Im Mystery-Land wimmelt es von Kobolden, Geistern und ähnlichen unangenehmen Zeitgenossen. Steuern Sie Ihre Spielfigur mit den Cursor-Steuertasten durch das Labyrinth. Der Computer wird Ihnen immer neue Aufgaben stellen, die Sie zu bestehen haben. Bei jedem Spiel ändern sich fast alle Variablen, so daß Sie immer neue Abenteuer erleben.
Kassette 19,80 DM

Programmbibliothek Nr. 13
MICRO FORTRAN
(für TRS 80, Video Genie)
Micro Fortran ist ein Fortran-System für den TRS-80/Video Genie mit mindestens 16 K RAM und benötigt **keine** Diskettenstation. Da Fortran eine sehr umfangreiche Sprache ist und der Micro Fortran schon ab 16 K RAM arbeiten soll, enthält Micro Fortran nicht alle Möglichkeiten von Fortran IV. Trotzdem versteht das System die wichtigsten Fortran-Befehle, beherrscht Realzahlenverarbeitung und hat einen bequemen, bildschirmorientierten Editor. Im Vergleich zu BASIC ist Fortran wesentlich schneller, strukturierte Programmierung mit Unterprogrammen ist einfacher usw. Nachteilig ist allerdings, daß das compilierte Programm zwar sehr viel schneller ist als ein BASIC-Programmm, aber dafür auch wesentlich mehr Speicherplatz verbraucht. Außerdem muß für Fortran immer der Quelltext UND das Objektprogramm im Speicher stehen.
Das gesamte Fortran-System einschließlich Editor und Laufzeitsystem benötigt knapp unter 8 K Byte, es bleibt dem Benutzer also selbst bei nur 16 K noch genügend Platz, um einfache Programme zu schreiben. Trotzdem sind 32 K oder mehr empfehlenswert.
Das Handbuch enthält eine Einführung in den Umgang mit FORTRAN und eine ausführliche Beschreibung aller unter MICRO FORTRAN verfügbaren Befehle.
Kassette und Handbuch 70,— DM

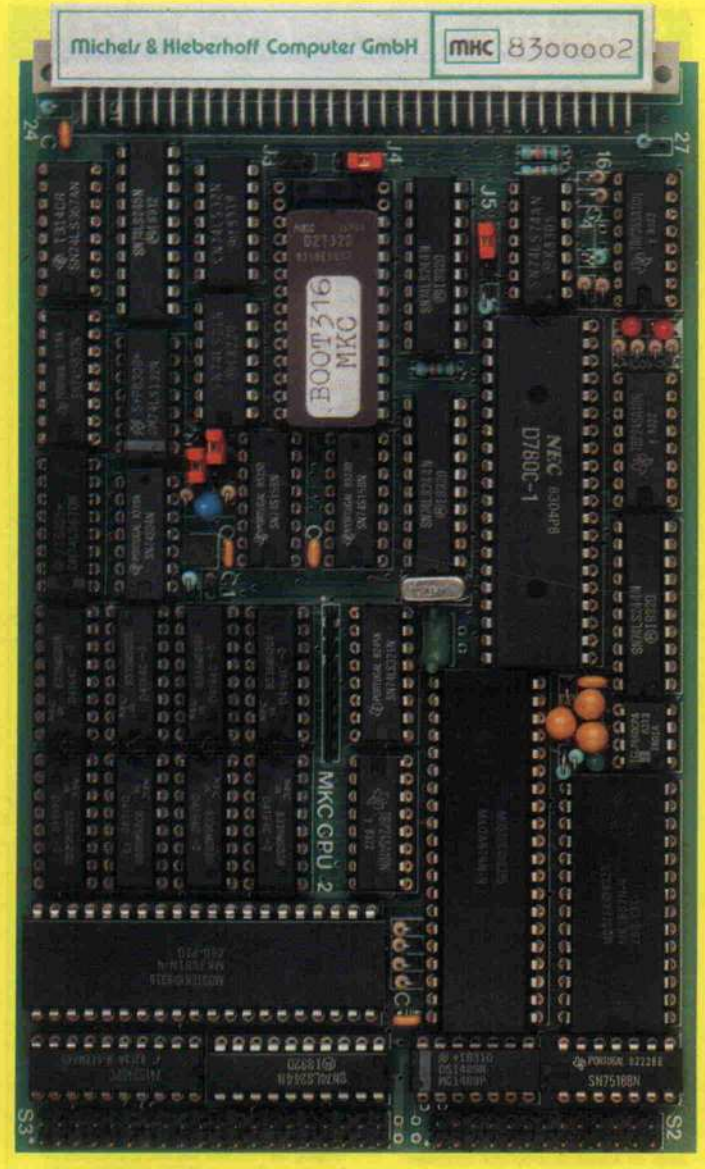
Programmbibliothek Nr. 14
OTHELLO
(für Apple mit Pascal)
Das Strategiespiel Othello (Reversi) in einer schnellen Pascal-Version. Drei Spielstärken sind einstellbar. Das Handbuch enthält das Listing mit sehr ausführlicher Beschreibung und ist deshalb besonders interessant für Pascal-Anfänger.
Diskette (5¼-Zoll) mit Handbuch 30,— DM

Neu:
CP/M 86 für
IBM PC 188,10 DM
(mit englischer Dokumentation)

So können Sie bestellen:
Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck oder einen von Ihrer Bank quittierten Einzahlungsbeleg über die Bestellsumme zuzüglich 3 DM (für Porto und Verpackung) bei. Bei Bestellung aus dem Ausland muß stets eine Überweisung in DM erfolgen. Die Überweisung und Ihre Bestellung richten Sie bitte an:

c't-Versand
Verlag Heinz Heise GmbH
Bissendorfer Straße 8
3000 Hannover 61
Konto-Nr. 93 05-308,
Postscheckamt Hannover

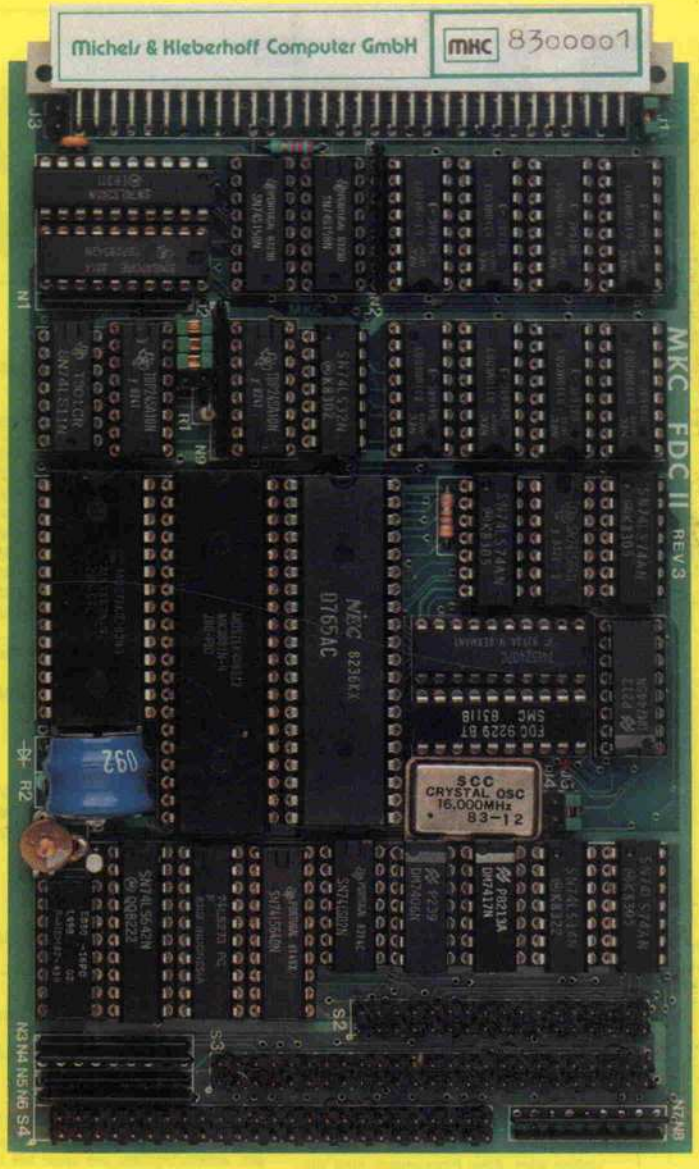
Das CP/M PLUS* SYSTEM



CPU II

Z 80-Zentraleinheit mit 64 KByte Speicher,
Adresserweiterung auf 1 MByte,
zwei seriellen und einer parallelen Schnittstelle.

898,– DM + MwSt = 1023,72 DM



FDC II

Floppy- und SASI-Controller für 5,25″ und 8″
Laufwerke (gleichzeitig!) mit eigenem
64 KByte Speicher und einer akkugepufferten Uhr.

1104,– DM + MwSt = 1258,56 DM

CP/M PLUS*

Das neue Betriebssystem für die CPU II und FDC II Karten. Implementiert ist die banked version für 128 K Speicher (erweiterbar), 16 I/O-Geräte, bis zu 8 Laufwerke (3 x 5,25″, 3 x 8″ Floppy- und 2 x 21 MByte-Winchester-Laufwerke. Alle Laufwerks- und Disketten-Parameter sind im Betrieb konfigurierbar. Ferner werden verschiedene fremde Diskettenformate automatisch erkannt und verarbeitet.

698,– DM + MwSt = 795,72 DM

Paketpreis:
2450,– DM + MwSt = 2793,– DM

*CPM PLUS ist ein eingetragenes Warenzeichen der Firma Digital Research.

Michels & Kleberhoff Computer GmbH MKC

Platzhoffstraße 11
5600 Wuppertal 1
Telefon 0202 / 308211

Händler:

DATAKAMP
Werwolf 4
5650 Solingen 1

GÜNTER STÖHR
Friedensstraße 22
5190 Stolberg
Tel. (02402) 73988

KRANICH GMBH
Frohnstraße 27
5620 Velbert 11
Tel. (02052) 2106

SEISSER
Diehlgasse 7
A-1050 Wien